# UTILITY PATENT APPLICATION TRANSMITTAL

*(Only for new nonprovisional applications under 37 CFR 1 53(b))*

| | |
|---|---|
| Attorney Docket No. | 200023US2RD |
| First Inventor or Application Identifier | Satoshi ITO |
| Title | ELECTRONIC CATALOG MAINTENANCE SYSTEM FOR ENABLING OUT-OF-STANDARD ELECTRONIC CATALOG CHANGES |

JC808 U.S. PTO 09/716947 11/22/00

JC760 U.S. PTO 11/22/00

## APPLICATION ELEMENTS

*See MPEP chapter 600 concerning utility patent application contents*

*ADDRESS TO:* Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ■ Fee Transmittal Form (e.g. PTO/SB/17) (Submit an original and a duplicate for fee processing)
2. ■ Specification — Total Sheets: 41
3. ■ Drawing(s) *(35 U.S.C. 113)* — Total Sheets: 34 (Formals)
4. ■ Oath or Declaration — Total Pages: 3
   a. ■ Newly executed (original)
   b. □ Copy from a prior application (37 C.F.R, §1.63(d)) (for continuation / divisional w/ box 16 completed)
      i. □ DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see *37 C.F.R. §1.63(d)(2) and 1.33(b).*
5. □ CD-ROM or CD-R in duplicate, large table or Computer Program *(Appendix)*
6. □ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. □ Computer Readable Form (CRF)
   b. Specification or Sequence Listing on:
      i. □ CD-ROM or CD-R (2 copies); or
      ii. □ Paper
   c. □ Statements verifying identity of above copies

## ACCOMPANYING APPLICATION PARTS

7. □ Assignment Papers (cover sheet & document(s))
8. □ Application Data Sheet. See 37 CFR 1.76
9. □ 37 C.F.R. §3.73(b) Statement *(when there is an assignee)* □ Power of Attorney
10. □ English Translation Document *(if applicable)*
11. □ Information Disclosure Statement (IDS)/PTO-1449 □ Copies of IDS Citations
12. □ Preliminary Amendment
13. ■ White Advance Serial No. Postcard
14. ■ Certified Copy of Priority Document(s) (1) *(if foreign priority is claimed)*
15. □ Applicant claims small entity status. *See 37 CFR 1.27*
16. ■ Other: Notice of Priority

16. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below:*

□ Continuation □ Divisional □ Continuation-in-part (CIP) of prior application no.:

*Prior application information:* Examiner: Group Art Unit:

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. Amend the specification by inserting before the first line the sentence:

□ This application is a □ Continuation □ Division □ Continuation-in-part (CIP) of application Serial No. Filed on

□ This application claims priority of provisional application Serial No. Filed

## 18. CORRESPONDENCE ADDRESS

22850

(703) 413-3000
FACSIMILE: (703) 413-2220

| | | | |
|---|---|---|---|
| Name: | Marvin J. Spivak | Registration No.: | 24,913 |
| Signature: | | Date: | 11/22/00 |
| Name: | C. Irvin McClellan Registration Number 21,124 | Registration No.: | |

Docket No. 200023US2RD

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

INVENTOR(S) Satoshi ITO

SERIAL NO: New Application

FILING DATE: Herewith

FOR: ELECTRONIC CATALOG MAINTENANCE SYSTEM FOR ENABLING OUT-OF-STANDARD ELECTRONIC CATALOG CHANGES

## FEE TRANSMITTAL

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

| FOR | NUMBER FILED | NUMBER EXTRA | RATE | CALCULATIONS |
|---|---|---|---|---|
| TOTAL CLAIMS | 24 - 20 = | 4 | × $18 = | $72.00 |
| INDEPENDENT CLAIMS | 3 - 3 = | 0 | × $80 = | $0.00 |
| ☐ MULTIPLE DEPENDENT CLAIMS (If applicable) | | | + $270 = | $0.00 |
| ☐ LATE FILING OF DECLARATION | | | + $130 = | $0.00 |
| | | | BASIC FEE | $710.00 |
| | | | TOTAL OF ABOVE CALCULATIONS | $782.00 |
| ☐ REDUCTION BY 50% FOR FILING BY SMALL ENTITY | | | | $0.00 |
| ☐ FILING IN NON-ENGLISH LANGUAGE | | | + $130 = | $0.00 |
| ☐ RECORDATION OF ASSIGNMENT | | | + $40 = | $0.00 |
| | | | TOTAL | $782.00 |

☐ Please charge Deposit Account No. 15-0030 in the amount of A duplicate copy of this sheet is enclosed.

■ A check in the amount of **$782.00** to cover the filing fee is enclosed.

■ The Commissioner is hereby authorized to charge any additional fees which may be required for the papers being filed herewith and for which no check is enclosed herewith, or credit any overpayment to Deposit Account No. 15-0030. A duplicate copy of this sheet is enclosed.

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Date: 11/22/00

Marvin J. Spivak
Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124

22850
Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 10/00)

# ELECTRONIC CATALOG MAINTENANCE SYSTEM FOR ENABLING OUT-OF-STANDARD ELECTRONIC CATALOG CHANGES

## BACKGROUND OF THE INVENTION

### FIELD OF THE INVENTION

The present invention relates to an electronic catalog maintenance system for maintaining an electronic catalog formed according to a prescribed standard such as ISO13584.

### DESCRIPTION OF THE BACKGROUND ART

ISO13584 (Parts Library) is the conventionally known international standard for implementing the electronic catalog system for electronically providing product information on the Internet. In this ISO13584, the electronic catalog is formed by dictionary data and contents (catalog data), which are given in a unified data structure in order to realize sharing and reuse of the product information.

In the dictionary data defined by ISO13584, the product classification is expressed hierarchically as in an example shown in Fig. 27, by relating "product classes" A0, B0, B1 and C0-C3 by a single tree structure. Each of the "product classes" A0, B0, B1, C0-C3 has "attribute items" V0-V6, and the "attribute items" of one "product class" (such as V0 and V1 belonging to B0, for example) are inherited to the lower "product classes" (C0 and C1).

Also, in ISO13584, a unique ID (identifier) called "BSU (Basic Semantic Unit) code" is to be assigned to each one of the "product classes" and the "attribute items" in order to identify it uniquely. Note that the "attribute items" includes "Visible" type attribute items which can only be referred and "Applicable" type attribute items for which actual values can be entered. The "Applicable" type

attribute items are to be selected from "Visible" type attribute items that can be referred (V0 of C2, for example).

While ISO13584 provides a framework for the electronic catalog, the international standardization of actual dictionary data is also in progress, and IEC61360 is currently striving for the standardization of the upper hierarchy part of the dictionary data in the field of electric and electronic products, that is, the general part regarding the "product classes" and the "attribute items". As a result, a product catalog producer of each company can creates that company's own contents by determining the original detailed "product classes" and "attribute items" as a lower level development from IEC61360.

A user of the electronic catalog can retrieve a desired product from the contents created in this way by tracing the classification hierarchy of the "product classes" and narrowing down products that can meet his/her needs by referring to attribute values. In recent years, several systems in compliance with ISO13584 have been developed in this trend.

In ISO13584 described above, a basic idea of the maintenance regarding a dictionary system formed according to the definition is described, and in particular, a mechanism based on Version/Revision updates is described for management of the dictionary.

Figs. 28A and 28B summarize that basic idea. In Fig. 28A, a way of handling each type is described for addition, change, and deletion of associated information of the attribute item (property). Also, in Fig. 28B, a way of handling each type is described for addition, change, and deletion of associated information of the product class (which will be referred to hereafter as "class").

However, according to the agreement shown in Figs. 28A and 28B, changes of the dictionary are very limited, and in

particular, changes required in the following cases cannot be handled.

Case 1) Deletion of Visible/Applicable Property

Case 2) Change of Visible/Applicable Property

Case 3) Change which deletes a Property to be inherited, among changes of Super Class

Case 4) Change of Name scope of Visible Property

Note however that, in Case 2), the "Version UP of Property" event is excluded, and addition of Visible/Applicable Property can be handled by Version up.

Note also that, in Case 3), the "Version UP of Super Class" event is excluded, and change which does not delete a Property to be inherited such as insertion of an intermediate class can be handled by Version up, while change of Preferred Name can be handled by Revision up.

Now, it is expected that these Case 1) to Case 4) will occur frequently in practice as in the case of changing the class hierarchy structure. For example, they are expected to occur in the case of newly creating a class (FFF) at an end (EEE) of the tree structure as shown in Fig. 29, the case of deleting an end class (HHH) as shown in Fig. 30, the case of merging a plurality of product classes (EEE and HHH) to one product class (KKK) as shown in Fig. 31, the case of moving the product class (HHH or DDD) as shown in Figs. 32, 33, 34 and 35, and the case of creating or deleting an intermediate class (HHH) as shown in Figs. 36 and 37.

Consequently, it is practically very important to enable handling of such cases as Case 1) to Case 4) by going beyond the framework of ISO13584. However, the following problems are encountered in realizing this.

First, the BSU code that is once issued and disclosed cannot be deleted and must be permanently managed even when it becomes unused as it is taken out from the dictionary system, because there is a possibility of referencing from

the legacy system, so that there is a need to be compatible with the legacy system.

Second, one class or attribute can be regarded as a number of different classes or attributes when the hierarchy structure is changed, even if the definition and the content of that class or attribute as a single entity remain unchanged. This will cause a large number of the new issuance of the BSU code and the releasing of the BSU code so that the management of codes becomes very difficult. For this reason, there is a need to provide a function for comprehending the new issuance of the BSU code.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide an electronic catalog maintenance system capable of improving usefulness and generality of the electronic catalog by ensuring the freedom of changes of the electronic catalog, while making the changed electronic catalog data utilizable even in the conventional systems.

According to one aspect of the present invention there is provided an electronic catalog maintenance system, comprising: a dictionary database configured to store dictionary data of an electronic catalog, the dictionary data being given in a form of a tree structure formed by identifiers for uniquely identifying classes classifying products and attributes of the products; an editing unit configured to edit the dictionary data stored by the dictionary database by making changes including standard changes defined by a prescribed standard and out-of-standard changes not defined by the prescribed standard; a change status detection unit configured to detect a status of each change made by the editing unit, and to generate an identifier change data indicating the status of each change

made by the editing unit and updates of identifiers to be made in the distrionary data; an identifier update unit configured to issue a new identifier of each class or attribute created by an out-of-standard change made by the editing unit, and to retire an old identifier of each class or attribute deleted by an out-of-standard change made by the editing unit, according to the identifier change data generated by the change status detection unit; and an identifier change database configured to store the identifier change data generated by the change status detection unit.

According to another aspect of the present invention there is provided an electronic catalog maintenance method, comprising the steps of: (a) storing dictionary data of an electronic catalog in a dictionary database, the dictionary data being given in a form of a tree structure formed by identifiers for uniquely identifying classes classifying products and attributes of the products; (b) editing the dictionary data stored by the dictionary database by making changes including standard changes defined by a prescribed standard and out-of-standard changes not defined by the prescribed standard; (c) detecting a status of each change made by the step (b), and generating an identifier change data indicating the status of each change made by the step (b) and updates of identifiers to be made in the dictionary data; (d) issuing a new identifier of each class or attribute created by an out-of-standard change made by the step (b), and releasing an old identifier of each class or attribute deleted by an out-of-standard change made by the step (b), according to the identifier change data generated at the step (c); and (e) storing the identifier change data generated by the step (c) in an identifier change database.

According to another aspect of the present invention there is provided a computer usable medium having computer readable program codes embodied therein for causing a

computer to function as an electronic catalog maintenance system, the computer readable program codes include: a first computer readable program code for causing said computer to store dictionary data of an electronic catalog in a dictionary database, the dictionary data being given in a form of a tree structure formed by identifiers for uniquely identifying classes classifying products and attributes of the products; a second computer readable program code for causing said computer to edit the dictionary data stored by the dictionary database by making changes including standard changes defined by a prescribed standard and out-of-standard changes not defined by the prescribed standard; a third computer readable program code for causing said computer to detect a status of each change made by the editing unit, and to generate an identifier change data indicating the status of each change made by the editing unit and updates of identifiers to be made in the dictionary data; a fourth computer readable program code for causing said computer to issue a new identifier of each class or attribute created by an out-of-standard change made by the editing unit, and to retire an old identifier of each class or attribute deleted by an out-of-standard change made by the editing unit, according to the identifier change data generated by the third computer readable program code; and a fifth computer readable program code for causing said computer to store the identifier change data generated by the third computer readable program code in an identifier change database.

Other features and advantages of the present invention will become apparent from the following description taken in conjunction with the accompanying drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing an overall configuration of an electronic catalog maintenance system according to one embodiment of the present invention.

Fig. 2 is a diagram showing an example of class data in electronic catalog dictionary data used by the electronic catalog maintenance system of Fig. 1.

Fig. 3 is a diagram showing an example of attribute data in electronic catalog dictionary data used by the electronic catalog maintenance system of Fig. 1.

Fig. 4 is a diagram showing an example of class-attribute relationship data in electronic catalog dictionary data used by the electronic catalog maintenance system of Fig. 1.

Fig. 5 is a diagram showing an example of class BSU code change data in BSU code change data used by the electronic catalog maintenance system of Fig. 1.

Fig. 6 is a diagram showing an example of attribute BSU code change data in BSU code change data used by the electronic catalog maintenance system of Fig. 1.

Fig. 7 is a diagram showing an exemplary model for expressing version tree data and revision tree data used by the electronic catalog maintenance system of Fig. 1.

Fig. 8 is a diagram showing an exemplary data description of version tree data and revision tree data used by the electronic catalog maintenance system of Fig. 1.

Fig. 9 is a diagram showing an example of dictionary system quality data used by the electronic catalog maintenance system of Fig. 1.

Fig. 10 is a flow chart for an overall processing of the electronic catalog maintenance system of Fig. 1.

Fig. 11 is a flow chart for a processing by a master database management unit at the step S3 of Fig. 10.

Fig. 12 is a flow chart for a processing by an editing database management unit at the step S4 of Fig. 10.

Fig. 13 is a flow chart for a processing by a dictionary change status detection function at the step S13 of Fig. 10.

Fig. 14 is a diagram showing an example of change type discrimination rules used by the electronic catalog maintenance system of Fig. 1.

Fig. 15 is a flow chart for a processing by a summary generation function with respect to class data at the step S16 of Fig. 10.

Fig. 16 is a flow chart for a processing by a summary generation function with respect to attribute data at the step S16 of Fig. 10.

Fig. 17 is a diagram showing an exemplary manner of changing BSU codes due to dictionary data editing at the step S12 of Fig. 10.

Fig. 18 is a diagram showing class BSU code change data and its summary generated as a result of changes shown in Fig. 17.

Fig. 19 is a diagram showing attribute BSU code change data and its summary generated as a result of changes shown in Fig. 17.

Fig. 20 is a flow chart for a processing by a dictionary system quality check function at the step S17 of Fig. 10.

Fig. 21A is a diagram showing an example of quality check rules used by the electronic catalog maintenance system of Fig. 1.

Fig. 21B is a diagram showing an example of BSU code issuance rules used by the electronic catalog maintenance system of Fig. 1.

Fig. 22 is a flow chart for a processing by an editing database management unit at the step S18 of Fig. 10.

Fig. 23 is a flow chart for a processing by a BSU code update function with respect to class data at the step S22 of Fig. 10.

Fig. 24 is a flow chart for a processing by a BSU code update function with respect to attribute data at the step S22 of Fig. 10.

Fig. 25 is a flow chart for a processing by a master database management unit at the step S23 of Fig. 10.

Fig. 26 is a diagram showing computer readable recording media for recording an electronic catalog maintenance program according to the present invention.

Fig. 27 is a diagram showing an exemplary electronic catalog data structure according to ISO13584.

Fig. 28A is a diagram showing Version/Revision update rules for property (attribute) as defined by ISO13584.

Fig. 28B is a diagram showing Version/Revision update rules for class (attribute) as defined by ISO13584.

Fig. 29 is a diagram showing an exemplary BSU code change due to an electronic catalog change for creation of end class.

Fig. 30 is a diagram showing an exemplary BSU code change due to an electronic catalog change for deletion of end class.

Fig. 31 is a diagram showing an exemplary BSU code change due to an electronic catalog change for merging of classes.

Fig. 32 is a diagram showing an exemplary BSU code change due to an electronic catalog change for moving of a class.

Fig. 33 is a diagram showing an exemplary BSU code change due to an electronic catalog change for moving of a class.

Fig. 34 is a diagram showing an exemplary BSU code change due to an electronic catalog change for moving of a class.

Fig. 35 is a diagram showing an exemplary BSU code change due to an electronic catalog change for moving of a class.

Fig. 36 is a diagram showing an exemplary BSU code change due to an electronic catalog change for creation of an intermediate class.

Fig. 37 is a diagram showing an exemplary BSU code change due to an electronic catalog change for deletion of an intermediate class.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to Fig. 1 to Fig. 26, one embodiment of the electronic catalog maintenance system according to the present invention will be described in detail.

(Overall configuration of the electronic catalog maintenance system)

Fig. 1 shows a schematic configuration of the electronic catalog maintenance system according to this embodiment.

As shown in Fig. 1, the electronic catalog maintenance system of this embodiment comprises a dictionary editor 1, a BSU code change management function 2, a dictionary change status detection function 5, a dictionary system quality check function 6, an editing database management unit 7, a master database management unit 8, an electronic catalog dictionary editing database 14, a BSU code change management editing database 16, an electronic catalog dictionary master database 20, and a BSU code change management master database 22.

The dictionary editor 1 newly creates or edits the electronic catalog dictionary data 10. In this embodiment, this dictionary editor 1 can store the newly created or edited electronic catalog dictionary data 10 into the electronic catalog dictionary editing database 14 or the electronic catalog dictionary master database 20, and read out the electronic catalog dictionary data 10 from the

electronic catalog dictionary editing database 14 or the electronic catalog dictionary master database 20 and edit them.

The BSU code change management function 2 has a summary generation function 3 and a BSU code update function 4. The summary generation function 3 generates a summary of changes by deleting redundant portion from the BSU code change data 9 at a time of storing the electronic catalog dictionary data 10. The BSU code update function 4 updates the BSU codes in the electronic cataqlog dictionary data 10 according to the generated summary and BSU code issuance rules 11.

The dictionary change status detection function 5 detects how BSU codes will be affected by the editing of the electronic catalog dictionary data 10 made by the dictionary editor 1 according to change type discrimination rules 12, and generates the BSU code change data 9.

The dictionary system quality check function 6 checks a quality of the updated electronic catalog dictionary data 10 according to quality check rules 13, and generates dictionary system quality data 26.

The editing database management unit 7 manages data input/output with respect to the electronic catalog dictionary editing database 14 and the BSU code change management editing database 16.

The master database management unit 8 manages data input/output with respect to the electronic catalog dictionary master database 20 and the BSU code change management master database 22.

The electronic catalog dictionary data 10 updated as a result of the editing or the like will be stored into the electronic catalog dictionary editing database 14 by the editing database management unit 7, as editing electronic catalog dictionary data 15. Also, the BSU code change data 9 that became the summary and the dictionary system quality

data 26 are stored into the BSU code change management editing database 16 by the editing database management unit 7 similarly, as editing BSU code change data 17 and editing dictionary system quality data 19 respectively. At a time of storing these data, the editing database management unit 7 generates and manages Revision Tree data 18 describing relationships among these data in order to carry out the revision management of the dictionary.

Among the electronic catalog dictionary data 15 stored in the electronic catalog dictionary editing database 14, data to be used as a disclosure version which have a high level of completeness will be sent to the master database management unit 8, and managed in the electronic catalog dictionary master database 20 as master electronic catalog dictionary data 21. At the same time, the corresponding editing BSU code change data 17 and the corresponding editing dictionary system quality data 19 will be also stored in the BSU code change management master database 22 as master BSU code change data 23 and master dictionary system quality data 25 respectively. At a time of storing these data, the master database management unit 8 generates and manages Version Tree data 24 describing relationships among these data in order to carry out the version management of the dictionary.

(Configuration of the electronic catalog dictionary data)

Next, the contents of the electronic catalog dictionary data 10 or 15 to be stored in the electronic catalog dictionary editing database 14 and edited by the dictionary editor 1 will be described. Fig. 2 to Fig. 4 show exemplary configurations of the electronic catalog dictionary data 10 or 15.

Fig. 2 shows an example of class data in the electronic catalog dictionary data 10 or 15, where one line indicates information of one class. In this information, a

"CID" which is an ID for identifying the class internally, a "BSU" which describes a BSU code of the class, and a "parent class CID" which describes a CID of a class that is a parent on the hierarchy structure of classes will be automatically assigned by the dictionary editor 1 and the BSU code change management function 2 in this embodiment. Note that the electronic catalog dictionary data 10 or 15 in this embodiment contain associated information of the class as specified by ISO13584 including a "Preferred Name" which indicates a name of the class, and this will be entered by a user. Also, a value of a "data quality level" in this electronic catalog dictionary data 10 or 15 will be written by the dictionary system quality check function 6.

Fig. 3 shows an example of attribute data in the electronic catalog dictionary data 10 or 15, where one line indicates information of one attribute. In this information, a "PID" which is an ID for identifying the attribute internally and a "BSU" which describes a BSU code of the attribute will be automatically assigned by the dictionary editor 1 and the BSU code change management function 2 in this embodiment. There is also associated information of the attribute as specified by ISO13584 including a "Preferred Name" which indicates a name of the attribute, and this will be entered by a user. Also, a value of the data quality level will be written by the dictionary system quality check function 6.

Fig. 4 shows an example of class-attribute relationship data in the electronic catalog dictionary data 10 or 15, where one line indicates information of one class-attribute relationship. In this information, a "CID_scope" describes a class for which this attribute is defined, a "Visible" describes a list of classes that can be referred by this attribute, and "Applicable" describes a list of classes to which this attribute can be applied.

(Configuration of the BSU code change data)

Next, the contents of the BSU code change data 9 or 17 to be generated by the editing of the electronic catalog dictionary data 10 or 15 by the dictionary editor 1 and stored in the BSU code change management editing database 16 will be described. Fig. 5 and Fig. 6 show exemplary configurations of the BSU code change data 9 or 17.

Fig. 5 shows an example of BSU code change data for classes in the BSU code change data 9 or 17 where one line indicates information of one BSU code change event. In this information, a "Status" indicates an event type which can take four values (VUP, RUP, NEW, OOD). Here, "VUP" and "RUP" indicate change events in accordance with the Version/Revision management specification of ISO13584 as shown in Figs. 28A and 28B. Also, "NEW" indicates one for which there is a need to newly issue a BSU code as a result of a change event. Also, "OOD" indicates one for which there is a need to retire a BSU code as it is excluded from the dictionary system.

For each "Status", "CID" of the class with respect to which the change is made, "BSU", and "Refer_to" and "Same_as" which indicate relationships of generation changes among classes are given. Using these two relationships, it is possible to access the registered class through the generation change, even in the case of making access to the dictionary by using a retired BSU code which is already excluded from the dictionary system.

Fig. 6 shows an example of BSU code change data for attributes in the BSU code change data 9 or 17, where one line indicates information of one BSU code change event. In this information, a "Status" indicates an event type which can take four values (VUP, RUP, NEW, OOD). Here, "VUP" and "RUP" indicate change events in accordance with the Version/Revision management specification of ISO13584 as shown in Figs. 28A and 28B. Also, "NEW" indicates one for which there is a need to newly issue a BSU code as a result

of a change event. Also, "OOD" indicates one for which there is a need to retire a BSU code as it is excluded from the dictionary system.

For each "Status", "PID" of the attribute with respect to which the change is made, "BSU", and "Refer_to" and "Same_as" which indicate relationships of generation changes among attributes are given. Using these two relationships, it is possible to access the registered attribute through the generation change, even in the case of making access to the dictionary by using a retired BSU code which is already excluded from the dictionary system.

(Configuration of Version Tree data)

Next, the Version Tree data to be stored in the BSU code change management editing database 16 or the BSU code change management master database 22 will be described.

Fig. 7 shows a model for expressing the Version Tree data 24 and the Revision Tree data 18 used in Fig. 1 in terms of EXPRESS-G, which is an ER model in which a rectangle represents an entity and a line represents a relation.

In Fig. 7, a "DB_Version" entity represents the master electronic catalog dictionary data 21, a "Version_History" entity represents the master BSU code change data 23, a "DB_Revision" entity represents the editing electronic catalog dictionary data 10, a "Revision_History" entity represents the editing BSU code change data 17, and a "DB_quality" entity represents the editing dictionary system quality data 19 and the master dictionary system quality data 25.

Fig. 8 shows an exemplary data description of the Version Tree data 24 and the Revision Tree data 18, which is described according to a model shown in Fig. 7, and which will be used as information for the version management of each data. In Fig. 8, a portion enclosed by an enclosure 30 represents the Version Tree data 24, and a

portion enclosed by an enclosure 31 represents the Revision Tree data 18.

(Configuration of the dictionary system quality data)

Next, the contents of the dictionary system quality data 26 to be generated by the dictionary system quality check function 6 will be described.

Fig. 9 shows an example of the dictionary system quality data 26, where quality evaluation values are described from viewpoints of classes, attributes, and the entire dictionary itself. These values are calculated by the dictionary system quality check function 6 by evaluating the electronic catalog dictionary data 10 and the BSU code change data 9 according to the quality check rules 13.

(Processing by the electronic catalog maintenance system)

The electronic catalog maintenance system of this embodiment in the above described configuration carries out the electronic catalog maintenance operation as follows. Fig. 10 shows a flow chart for an overall processing to be carried out by the electronic catalog maintenance system of this embodiment.

As shown in Fig. 10, when the dictionary editor 1 is activated (step S1), whether the electronic catalog dictionary data to be edited is a new one or an existing one is judged (entered) (step S2). Then, in the case of newly creating an electronic catalog dictionary, the processing for generating a new dictionary is carried out at the steps S3 and S4, whereas in the case of reading existing data, whether that data is stored in the electronic catalog dictionary editing database 14 or the electronic catalog dictionary master database 20 is judged (step S5). In the case of reading from the electronic catalog dictionary editing database 14, the processing of the step S11 is carried out, whereas in the case of reading

from the electronic catalog dictionary master database 20, the processing of the steps S6 to S10 is carried out.

Next, the electronic catalog dictionary data 10 so created or read will be changed by the editing processing of the step S12. Then, the content of that change is checked by the processing of the step S13. Namely, the dictionary change status detection function 5 compares the editing content made by the dictionary editor 1 with the change type discrimination rules 12 to determine the editing content as one of a Revision UP change, a Version UP change and a New BSU change, and generates the check result as the BSU code change data 9.

When it is desired to store the edited electronic catalog dictionary data 10 (step S14 YES and step S15 YES), it is stored into the electronic catalog dictionary editing database 14 (step S18) after the summary generation processing of the step S16 and the quality check processing of the step S17. On the other hand, when it is desired to store the electronic catalog dictionary data 10 with a high level of completeness as the master data (step S19 YES), it is stored into the electronic catalog dictionary master database 20 (step S23) after the quality of the dictionary is confirmed at the steps S20 and S21, and the BSU code is issued and the Version/Revision update is carried out at the step S22.

In the following, each processing will be described in further detail. Fig. 11 shows a flow chart for a processing to be carried out by the master database management unit 8 in the processing of the step S3 of Fig. 10, that is the processing for newly creating the electronic catalog dictionary data 10.

As shown in Fig. 11, in the processing of the step S3 described above, when the master database (name) is entered at the step S30, the Version Tree data 24 based on a model shown in Fig. 7 is created and stored by the steps S31 to

S34, and the necessary information is transmitted to the editing database management unit 7 by the step S35. Then, an empty electronic catalog dictionary data 21 is created and stored by the step S36 and an empty BSU code change data 23 is created and stored by the step S37.

Fig. 12 shows a flow chart for the processing of the step S4 in Fig. 10, that is a processing to be carried out by the editing database management unit 7 in the processing for newly creating the electronic catalog dictionary data 10.

As shown in Fig. 12, the necessary information is received from the master database management unit 8 at the step S40, and the Revision Tree data 18 is created and stored by the steps S41 to S45. Then, an empty electronic catalog dictionary data 15 is created and stored by the step S46 and an empty BSU code change data 17 is created and stored by the step S47. At the same time, an empty electronic catalog dictionary data 10 is created and stored by the step S48 and an empty BSU code change data 9 is created and stored by the step S49.

Fig. 13 shows a flow chart for the processing of the step S13 in Fig. 10, that is a processing to be carried out by the dictionary change status detection function 5 in the processing for checking the change content of the edited electronic catalog dictionary data 10.

As shown in Fig. 13, the dictionary change event is detected at the step S100 and discriminated according to the change type discrimination rules 12 shown in Fig. 14 at the step S101. Then, depending on whether the event type is that related to the class or that related to the attribute (step S102), the subsequent steps S103 to S110 are carried out when the event type is that related to the class, whereas the subsequent steps S111 to S118 are carried out when the event type is that related to the attribute, so as to generate the BSU code change data 9.

Fig. 14 shows an example of the change type discrimination rules 12 describing the rules to be used in the processing of Fig. 13.

As shown in Fig. 14, the change type discrimination rules 12 are largely divided into rules for the product class change and the rules for the attribute change, and each rule describes a condition for judging whether each change is Revision UP, Version UP or New BSU, and a processing method suitable for each type. Note that these judgement condition and processing method are described in the IF-THEN format in this embodiment, but they may be described in any other format.

Then, as a result of the judgement according to the change type discrimination rules 12, if the change of the BSU code change data 9 is related to the class, whether that change is Revision UP, Version UP, or New BSU is checked, whether there is a class to be retired or not is checked, and the BSU code change data 9 is changed according to the rule to be applied. If the change of the BSU code change data 9 is related to the attribute, the rule to be applied is checked similarly.

Fig. 15 shows a flow chart for a processing to be carried out by the summary generation function 3 with respect to the class data in the processing of the step S16 of Fig. 10. Here, the processing is branched according to "Status" of each event described in the BSU code change data 9 and the status of BSU code issuance. By this processing, the redundant operation can be removed from the editing operation, and only significant changes including the BSU code change can be extracted.

Among symbols used in Fig. 15, "Status" indicates the "Status" attribute of the class BSU code change data, "BSU" indicates the "BSU" attribute of the class BSU code change data, "CID" indicates the "CID" attribute of the class BSU code change data, "Same_as" indicates "Same_as" attribute

of the class BSU code change data, "Refer_to" indicates the "Refer_to" attribute of the class BSU code change data, and "A.Status" indicates the "Status" attribute value of the class BSU code change data A.

Also, among alphabets attached to these symbols, "R" indicates the tail end data of the class BSU code change data, "X" indicates the class BSU code change data whose CID value is equal to "R.Refer_to" value, "S" indicates the class BSU code change data whose CID value is that of the parent (upper level) class of "R.CID" value and whose "Status" value is "NEW", "Y" indicates the class BSU code change data whose "Status" value is "NEW" and whose CID value is equal to "R.CID" value, "Z" indicates the class BSU code change data whose "Status" value is "OOD" and whose "Refer_to" value is equal to "Y.CID" value, "T" indicates the class BSU code change data whose "Status" value is "VUP" or "RUP" and whose CID value is equal to "R.CID" value, and "K" indicates the class BSU code change data whose "Status" value is equal to "R.Status" value and whose CID value is equal to "R.CID" value.

First, the BSU code change data related to the class is read (step S201), the tail end data R is detected (step S202), and the processing is terminated when the data R does not exist (step S203 NO), or otherwise the processing from the step S204 on is carried out.

In the processing from the step S204 on, whether "R.Status" value is "OOD" or not is judged (step S204), and if "R.Status" is "OOD", whether "R.BSU" value is "NULL" or not is judged (step S205).

When it is judged that "R.BSU" value is "NULL" at the step S205, Y for which "Y.Status" value is "NEW" and "Y.CID" value is equal to "R.CID" value is obtained, Z for which "Z.Status" value is "OOD" and "Z.Refer_to" value is equal to "Y.CID" value is obtained, and X for which "X.Status" value is "NEW" and "X.CID" value is equal to

"R.Refer_to" value is obtained (steps S209, S210, S211). Then, whether such X and Z exist or not is judged (steps S212, S213, S215). Then, depending on a combination of the existing X and Z, "NULL" is substituted into "Z.Refer_to" (step S214), "R.CID" is replaced by "Z.CID" and "X.CID" is substituted into "X.Refer_to" (steps S216, S217), "R.CID" is deleted (step S218), every T for which "T.Status" is "VUP" or "RUP" and "T.CID" value is equal to "R.CID" value is obtained (step S219), and R, Y and T are deleted (step S220).

When it is judged that "R.BSU" value is not "NULL" at the step S205, X for which "X.CID" value is equal to "R.Refer_to" value is obtained (step S206), and if such X does not exist (step S207 NO), the parent class S of "R.CID" is obtained and "S.CID" is set to "R.Refer_to" (step S208).

On the other hand, when it is judged that the "R.Status" value is not "OOD" at the step S204, whether "R.Status" value is "NEW" or not is judged (step S221), and if it is "NEW", every K for which "K.Status" value is equal to "R.Status" value and "K.CID" value is equal to "R.CID" value is obtained (step S222) and K is deleted (step S223). Then, whether "R.BSU" is "NULL" or not is judged (step S224), and if so R is deleted (step S225).

After R is obtained in this way, the data immediately before R is substituted into this obtained data R (step S226), and the next processing is started by the loop processing.

Fig. 16 shows a flow chart for a processing to be carried out by the summary generation function 3 with respect to the attribute in the processing of the step S16 of Fig. 10. Here, the processing is branched according to "Status" of each event described in the BSU code change data 9 and the status of BSU code issuance, similarly as Fig. 15. By this processing, the redundant operation such

as a change related to the class that was once newly created but then deleted on the second thought, for example, can be removed from the editing operation, and only significant changes including the BSU code change can be extracted.

Among symbols used in Fig. 16, "PID" indicates the "PID" attribute of the attribute BSU code change data, and "A.PID" indicates the "PID" attribute value of the class BSU code change data A. Also, among alphabets attached to these symbols, "R" indicates the tail end data of the attribute BSU code change data, "X" indicates the attribute BSU code change data whose "Status" value is "NEW" and whose PID value is equal to "R.Refer_to" value, "Y" indicates the attribute BSU code change data whose "Status" value is "NEW" and whose PID value is equal to "R.PID" value, "Z" indicates the attribute BSU code change data whose "Status" value is "OOD" and whose "Refer_to" value is equal to "Y.PID" value, "T" indicates the attribute BSU code change data whose "Status" value is "VUP" or "RUP" and whose PID value is equal to "R.PID" value, and "K" indicates the attribute BSU code change data whose "Status" value is equal to "R.Status" value and whose PID value is equal to "R.PID" value.

First, the BSU code change data related to the attribute is read (step S301), the tail end data R is detected (step S302), and the processing is terminated when the data R does not exist (step S303 NO), or otherwise the processing from the step S304 on is carried out.

In the processing from the step S304 on, whether "R.Status" value is "OOD" or not is judged (step S304), and if "R.Status" is "OOD", whether "R.BSU" value is "NULL" or not is judged (step S305).

When it is judged that "R.BSU" value is "NULL" at the step S305, Y for which "Y.Status" value is "NEW" and "Y.PID" value is equal to "R.PID" value is obtained, Z for

which "Z.Status" value is "OOD" and "Z.Refer_to" value is equal to "Y.PID" value is obtained, and X for which "X.Status" value is "NEW" and "X.PID" value is equal to "R.Refer_to" value is obtained (steps S306, S307, S308). Then, whether such X and Z exist or not is judged (steps S309, S310, S312). Then, depending on a combination of the existing X and Z, "NULL" is substituted into "Z.Refer_to" (step S311), "R.PID" is replaced by "Z.PID" and "X.PID" is substituted into "X.Refer_to" (steps S313, S314), "R.PID" is deleted (step S315), every T for which "T.Status" is "VUP" or "RUP" and "T.PID" value is equal to "R.PID" value is obtained (step S316), and R, Y and T are deleted (step S317).

On the other hand, when it is judged that the "R.Status" value is not "OOD" at the step S304, whether "R.Status" value is "NEW" or not is judged (step S318), and if it is "NEW", every K for which "K.Status" value is equal to "R.Status" value and "K.PID" value is equal to "R.PID" value is obtained (step S319) and K is deleted (step S320). Then, whether "R.BSU" is "NULL" or not is judged (step S321), and if so R is deleted (step S322).

After R is obtained in this way, the data immediately before R is substituted into this obtained data R (step S323), and the next processing is started by the loop processing. When it is judged that "R.BSU" value is not "NULL" at the step S305, and when it is judged that "R.Status" value is "NEW" at the step S318, this step S323 is carried out immediately.

(Change of the BSU code)

In the following, the concrete examples of the electronic catalog maintenance by the electronic catalog maintenance system described above will be described. Fig. 17 shows an exemplary BSU code change by the dictionary data editing processing of the step S12 of Fig. 109.

First, as shown in a part (a) of Fig. 17, suppose that

B1 is deleted and the note of C0 is changed. As a result, as shown in a part (b) of Fig. 17, C2 and C3 located below the deleted B1 are directly connected below A0 as C4 and C5, and Revision UP of C0 for which the note is changed is carried out. At this point, C2 and C3 located below the deleted B1 are set in retired states, and V4, V5 and V6 that are belonging only to B1, C2 and C3 are also set in retired states.

Next, when C4 and C5 are merged, as shown in a part (c) of Fig. 17, a new C6 is created instead of the merged C4 and C5, and C4 and C5 are set in retired states. At this point, V7, V8, V9 and V10 that are belonging only to the merged C4 and C5 are also set in retired states, and V11, V12, V13 and V14 are newly added instead of them.

(Summary generation)

In the case of making such a change in the electronic catalog dictionary data, the BSU code change data for the class and the attribute created by the change and their summary will be generated as follows.

A part (a) of Fig. 18 shows the class BSU code change data created by the dictionary change status detection function 5 when the operations of Fig. 17 are carried out. Then, by carrying out the processing of Fig. 15 described above with respect to these data, these data are updated as shown in parts (b) to (e) of Fig. 18. In this way, it can be seen that only the significant change operations for the class are extracted.

Fig. 19 shows the attribute BSU code change data created by the change of Fig. 17 and a manner of generating their summary.

A part (a) of Fig. 19 shows the attribute BSU code change data created by the dictionary change status detection function 5 when the operations of Fig. 17 are carried out. Then, by carrying out the processing of Fig. 16 described above with respect to these data, these data

are updated as shown in parts (b) to (i) of Fig. 19. In this way, it can be seen that only the significant change operations for the attribute are extracted.

(Quality check of the dictionary)

Next, the operation of the dictionary quality check function, which is one of the functions of the electronic catalog maintenance system of this embodiment, will be described. Fig. 20 shows a flow chart for a processing of the dictionary system quality check function 6 to generate the dictionary system quality data 26 in the processing of the step S17 of Fig. 10.

First, by the steps S401 to S407 of Fig. 20, the class BSU code change data that became the summary are read, a top data R is obtained, the quality of individual class is evaluated according to the quality check rules 13, and its value is written into the "data quality level" field of the class data.

Here, as the quality check rules 13, tables formed by the evaluation conditions and the evaluation functions as shown in Fig. 21A can be used, for example. As shown in Fig. 21A, the quality check rules 13 can be formed from the evaluation of individual class, the evaluation of individual attribute, the evaluation regarding the class hierarchy structure, the evaluation regarding the attribute overlaps, and the evaluation regarding the entire dictionary, for example. Note that the quality evaluation is made by calculating the quality level by using such quality check rules 13 in this embodiment, but it is also possible to account for the importance of associated items, and it is also possible to set evaluation functions based on different evaluation axes.

Then, at the steps S408 and S409, the overall quality of the class data such as the consistency of the hierarchy structure for example is evaluated according to the quality check rules 13, and added to the dictionary system quality

data 26. In addition, at the steps S410 to S416, the quality of individual attribute is evaluated according to the quality check rules 13 and its value is written into the "data quality level" field of the attribute data.

Next, at the steps S417 and S418, the overall quality of the attribute data such as the overlapping existence of identical definitions for example is evaluated according to the quality check rules 13, and added to the dictionary system quality data 26. Then, at the step S419, the dictionary system quality data 26 are evaluated according to the quality check rules 13, and the quality level of the entire electronic catalog dictionary is calculated and added to the dictionary system quality data 26.

(Processing by the editing database management unit 7)

The data produced by each processing described above are stored by the editing database management unit 7. Fig. 22 shows a flow chart for a processing of the editing database management unit 7 to store data in the processing of the step S18 of Fig. 10.

As shown in Fig. 22, at the steps S501 to S509, the DB_Revision entity and the Revision_History entity for describing relationships between the BSU code change data 9 and the electronic catalog dictionary data 10 that are desired to be stored in order to carry out the revision management of the dictionary are generated, and stored into the Revision Tree data 18.

Next, at the step S510 to S513, the quality information of the dictionary is obtained from the dictionary system quality data 26, and the DB_quality entity is generated and stored in the Revision Tree data 18 similarly. Then, at the step S514, the BSU code change data 9 created by the summary generation function 3 are stored into the BSU code change management editing database 16 as the editing BSU code change data 17. Finally, at the step S515, the electronic catalog dictionary data 10 is stored

into the electronic catalog dictionary editing database 14 as the editing electronic catalog dictionary data 15.

(Processing of BSU codes)

Next, the BSU code change management processing by the BSU code update function 4 will be described. Fig. 23 shows a flow chart for a processing by the BSU code update function 4 to update the class data in the processing of the step S22 of Fig. 10, and Fig. 24 shows a flow chart for a processing by the BSU code update function 4 to update the attribute data in the processing of the step S22 of Fig. 10.

In the case of the processing for updating the class data, as shown in Fig. 23, the processing is carried out according to the "Status" value of the BSU code change data 9 read by the step S601.

Namely, when it is judged that "Status" is "VUP" at the step S603, the Version UP is carried out at the steps S604 to S608. On the other hand, when it is judged that "Status" is not "VUP" at the step S603 and it is judged that "Status" is "RUP" at the step S609, the Revision UP is carried out at the steps S610 to S614. Else, when it is judged that "Status" is not "VUP" at the step S603 and it is judged that "Status" is not "RUP" at the step S609, whether "Status" is "NEW" or not is judged (step S615), and if "Status" is "NEW", the steps S616 to S618 are carried out according to the BSU code issuance rules 11 as shown in Fig. 21B, such that the new BSU code is issued and written into the BSU field of the class data.

Also, in the case of the processing for updating the attribute data, as shown in Fig. 24, the processing is carried out according to the "Status" value of the BSU code change data 9 read by the step S701.

Namely, when it is judged that "Status" is "VUP" at the step S703, the Version UP is carried out at the steps S704 and S705. On the other hand, when it is judged that

"Status" is not "VUP" at the step S703 and it is judged that "Status" is "RUP" at the step S706, the Revision UP is carried out at the steps S707 and S708. Else, when it is judged that "Status" is not "VUP" at the step S703 and it is judged that "Status" is not "RUP" at the step S706, whether "Status" is "NEW" or not is judged (step S709), and if "Status" is "NEW", the steps S710 to S712 are carried out according to the BSU code issuance rules 11 as shown in Fig. 21B, such that the new BSU code is issued and written into the BSU field of the attribute data.

(Processing by the master database management unit 8)

The electronic catalog dictionary data with a high level of completeness are stored into the electronic catalog dictionary master database 20 as the master electronic catalog dictionary data 21. Fig. 25 shows a flow chart for a processing of the master database management unit 8 to store data in the processing of the step S23 of Fig. 10.

As shown in Fig. 25, at the steps S801 to S806, the DB_Version entity and the Version_History entity for describing relationships between the BSU code change data 9 and the electronic catalog dictionary data 10 that are desired to be stored in order to carry out the version management of the dictionary are generated, and stored into the Version Tree data 24.

Next, at the step S807 to S810, the quality information of the dictionary is obtained from the dictionary system quality data 26, and the DB_quality entity is generated and stored in the Version Tree data 24 similarly. Then, at the step S811, the BSU code change data 9 created by the summary generation function 3 are stored into the BSU code change management master database 22 as the master BSU code change data 23. Finally, at the step S812, the electronic catalog dictionary data 10 is stored into the electronic catalog dictionary master database 20

as the master electronic catalog dictionary data 21.

(Electronic catalog maintenance program)

Note that the electronic catalog maintenance system described above can be realized by producing an electronic catalog maintenance program described in a prescribed programming language and installing this maintenance program into a general purpose computer such as PC, for example.

Namely, an electronic catalog maintenance software can be formed by a program for detecting the change status of the electronic catalog dictionary data 10 due to the editing by the dictionary editor 1 or the like and generating the BSU code change data 9 when the change that is out of the standard such as ISO13584 is made, and a program for recording the changed electronic catalog dictionary data 10 and the BSU code change data 9.

Note that in this electronic catalog maintenance software, it is preferable to include a program for generating the summary by simplifying the BSU code change data 9 by deleting the redundant portion from the change history of the BSU code change data 9 as described above.

This electronic catalog maintenance software may also include a program for setting the BSU codes used before the change in retired states when the change status is out of the standard such as ISO13584 and newly issuing corresponding codes to the catalog data or the dictionary data relevant to the out-of-standard change, and a program for generating and storing the BSU code change data 9 that records the correspondence between the retired BSU codes and the newly issued BSU codes.

This electronic catalog maintenance software may also include a program for evaluating the quality of the electronic catalog dictionary system and each element constituting the changed electronic catalog dictionary data 10 according to the quality check rules 13, generating the

dictionary system quality data 26, and storing the dictionary system quality data 26 into the BSU code change management editing database 16.

The electronic catalog maintenance software so produced can be recorded on recording media 116-119 that are readable by a general purpose computer 115 as shown in Fig. 26. More specifically, as shown in Fig. 26, this electronic catalog maintenance software can be recorded on computer readable recording media such as a magnetic recording medium like a floppy disk 116 or a cassette tape 119, an optical disk like a CD-ROM 117, and a RAM card 118.

Then, by using such computer readable recording media that record this electronic catalog maintenance software, it is possible to easily realize storing, carrying, and installing of a useful program that has an effect of enabling the efficient electronic catalog change management and ensuring generality of the electronic catalog by carrying out the comprehensive version management for the electronic catalog dictionary data relevant to the change operation.

As described, according to the electronic catalog maintenance system of the present invention, it becomes possible to enable the efficient electronic catalog change management and ensure generality of the electronic catalog by carrying out the comprehensive version management for the electronic catalog dictionary data relevant to the change operation such that the electronic catalog can be utilized without requiring a major modification to the existing systems.

Here, the prescribed electronic catalog standard can be any international standard such as ISO13584 (Parts Library) and IEC61360, for example. Also, the product classification information can be given by "product class (class)" or "attribute item (property)", for example. Also, the identifier can be given by BSU code, for example.

Also, the out-of-standard changes include addition or deletion of an intermediate hierarchy, issuance of new identifier, and a change of the tree structure that causes an occurrence of an identifier in a retired state. This change status can be detected according to Version/Revision updates, presence or absence of a new identifier (ID), and presence or absence of an identifier (ID) in a retired state, for example.

Thus, according to the present invention, it is possible to make the electronic catalog change management easier by comparing the electronic catalog before and after the change by utilizing the history information, and it is possible to improve degrees of freedom in the electronic catalog change operations by enabling the out-of-standard changes.

Also, according to the present invention, it is possible to make the change processing management easier by deleting unnecessary change processing (of the case where there is no change eventually or the case where there is only a minor change) such as those occurring in the case of the editing in which the changes are made repeatedly by trial and error, for example, from the history information.

Also, according to the present invention, it is possible to make access to the contents relevant to the out-of-standard change from a newly issued identifier, so that this system can be used in the legacy system and it is possible to ensure usefulness and generality of the electronic catalog.

Also, according to the present invention, it is possible to realize the change status management in accordance with the international standard such as ISO13584 (Parts Library) and IEC61360, for example, so that it is possible to improve generality of the electronic catalog.

Also, according to the present invention, it is possible to distinguish a minor change at a time of the

change operation and a formal change at a time of disclosure as a formal version up by selecting the storage unit according to the level of completeness of the electronic catalog, so that it is possible to provide the high quality dictionary.

Also according to the present invention, it is possible to improve the quality of the electronic catalog by evaluating the quality of the electronic catalog dictionary system and each element constituting the changed electronic catalog by using the dictionary system quality check function.

It is also to be noted that, besides those already mentioned above, many modifications and variations of the above embodiments may be made without departing from the novel and advantageous features of the present invention. Accordingly, all such modifications and variations are intended to be included within the scope of the appended claims.

WHAT IS CLAIMED IS:

1. An electronic catalog maintenance system, comprising:

a dictionary database configured to store dictionary data of an electronic catalog, the dictionary data being given in a form of a tree structure formed by identifiers for uniquely identifying classes classifying products and attributes of the products;

an editing unit configured to edit the dictionary data stored by the dictionary database by making changes including standard changes defined by a prescribed standard and out-of-standard changes not defined by the prescribed standard;

a change status detection unit configured to detect a status of each change made by the editing unit, and to generate an identifier change data indicating the status of each change made by the editing unit and updates of identifiers to be made in the distrionary data;

an identifier update unit configured to issue a new identifier of each class or attribute created by an out-of-standard change made by the editing unit, and to retire an old identifier of each class or attribute deleted by an out-of-standard change made by the editing unit, according to the identifier change data generated by the change status detection unit; and

an identifier change database configured to store the identifier change data generated by the change status detection unit.

2. The system of claim 1, wherein the change status detection unit generates the identifier change data that indicates the updates of identifiers to be made such that both new identifiers and old identifiers can be used in accessing the classes and the attributes after updating by the identifier update unit, by referring to the identifier

change data stored in the identifier change database.

3. The system of claim 1, wherein the editing unit makes the standard changes in forms of version/revision updates defined by ISO13584 standard, and the out-of-standard changes not defined by ISO13584.

4. The system of claim 1, wherein the change status detection unit detects the status of each change made by the editing unit according to occurrences of version/revision updates, issuance of the new identifier, and retirement of the old identifier.

5. The system of claim 1, further comprising:

a summary generation unit configured to simplify the identifier change data by deleting any redundant portion from the identifier change data generated by the change status detection unit;

wherein the identifier change database stores the identifier change data as simplified by the summary generation unit.

6. The system of claim 1, further comprising:

a quality check unit configured to generate a dictionary system quality data by evaluating qualities of the dictionary data as a whole and each element constituting the dictionary data according to prescribed quality check rules, after the dictionary data are edited by the editing unit;

wherein the identifier change database also stores the dictionary system quality data generated by the quality check unit.

7. The system of claim 1, wherein the dictionary database includes:

an editing dictionary database for storing a version of the dictionary data for which editing operations are in progress; and

a master dictionary database for storing a version of the dictionary data for which the editing operations are completed.

8. The system of claim 1, wherein the identifier change database includes:

an editing identifier change database for storing the identifier change data corresponding to a version of the dictionary data for which editing operations are in progress; and

a master identifier change database for storing the identifier change data corresponding to a version of the dictionary data for which the editing operations are completed.

9. An electronic catalog maintenance method, comprising the steps of:

(a) storing dictionary data of an electronic catalog in a dictionary database, the dictionary data being given in a form of a tree structure formed by identifiers for uniquely identifying classes classifying products and attributes of the products;

(b) editing the dictionary data stored by the dictionary database by making changes including standard changes defined by a prescribed standard and out-of-standard changes not defined by the prescribed standard;

(c) detecting a status of each change made by the step (b), and generating an identifier change data indicating the status of each change made by the step (b) and updates of identifiers to be made in the dictionary data;

(d) issuing a new identifier of each class or attribute created by an out-of-standard change made by the step (b),

and releasing an old identifier of each class or attribute deleted by an out-of-standard change made by the step (b), according to the identifier change data generated at the step (c); and

(e) storing the identifier change data generated by the step (c) in an identifier change database.

10. The method of claim 9, wherein the step (c) generates the identifier change data that indicates the updates of identifiers to be made such that both new identifiers and old identifiers can be used in accessing the classes and the attributes after updating by the step (d), by referring to the identifier change data stored in the identifier change database.

11. The method of claim 9, wherein the step (b) makes the standard changes in forms of version/revision updates defined by ISO13584 standard, and the out-of-standard changes not defined by ISO13584.

12. The method of claim 9, wherein the step (c) detects the status of each change made by the step (b) according to occurrences of version/revision updates, issuance of the new identifier, and retirement of the old identifier.

13. The method of claim 9, further comprising the step of:

(f) simplifying the identifier change data by deleting any redundant portion from the identifier change data generated by the step (c);

wherein the step (e) stores the identifier change data as simplified by the step (f) into the identifier change database.

14. The method of claim 9, further comprising the step of:

(g) generating a dictionary system quality data by

evaluating qualities of the dictionary data as a whole and each element constituting the dictionary data according to prescribed quality check rules, after the dictionary data are edited by the step (b);

wherein the step (e) also stores the dictionary system quality data generated by the step (g) in the identifier change database.

15. The method of claim 9, wherein the step (a) stores a version of the dictionary data for which editing operations are in progress, into an editing dictionary database of the dictionary database; and

the method further comprises the step of storing a version of the dictionary data for which the editing operations are completed, into a master dictionary database of the dictionary database.

16. The method of claim 9, wherein the step (e) stores the identifier change data corresponding to a version of the dictionary data for which editing operations are in progress, into an editing identifier change database of the identifier change database; and

the method further comprises the step of storing the identifier change data corresponding to a version of the dictionary data for which the editing operations are completed, into a master identifier change database of the identifier change database.

17. A computer usable medium having computer readable program codes embodied therein for causing a computer to function as an electronic catalog maintenance system, the computer readable program codes include:

a first computer readable program code for causing said computer to store dictionary data of an electronic catalog in a dictionary database, the dictionary data being

given in a form of a tree structure formed by identifiers for uniquely identifying classes classifying products and attributes of the products;

a second computer readable program code for causing said computer to edit the dictionary data stored by the dictionary database by making changes including standard changes defined by a prescribed standard and out-of-standard changes not defined by the prescribed standard;

a third computer readable program code for causing said computer to detect a status of each change made by the editing unit, and to generate an identifier change data indicating the status of each change made by the editing unit and updates of identifiers to be made in the dictionary data;

a fourth computer readable program code for causing said computer to issue a new identifier of each class or attribute created by an out-of-standard change made by the editing unit, and to retire an old identifier of each class or attribute deleted by an out-of-standard change made by the editing unit, according to the identifier change data generated by the third computer readable program code; and

a fifth computer readable program code for causing said computer to store the identifier change data generated by the third computer readable program code in an identifier change database.

18. The medium of claim 17, wherein the third computer readable program code generates the identifier change data that indicates the updates of identifiers to be made such that both new identifiers and old identifiers can be used in accessing the classes and the attributes after updating by the fourth computer readable program code, by referring to the identifier change data stored in the identifier change database.

19. The medium of claim 17, wherein the second computer readable program code makes the standard changes in forms of version/revision updates defined by ISO13584 standard, and the out-of-standard changes not defined by ISO13584.

20. The medium of claim 17, wherein the third computer readable program code detects the status of each change made by the second computer readable program code according to occurrences of version/revision updates, issuance of the new identifier, and retirement of the old identifier.

21. The medium of claim 17, further comprising:

a sixth computer readable program code for causing said computer to simplify the identifier change data by deleting any redundant portion from the identifier change data generated by the third computer readable program code;

wherein the fifth computer readable program code stores the identifier change data as simplified by the sixth computer readable program code in the identifier change database.

22. The medium of claim 17, further comprising:

a seventh computer readable program code for causing said computer to generate a dictionary system quality data by evaluating qualities of the dictionary data as a whole and each element constituting the dictionary data according to prescribed quality check rules, after the dictionary data are edited by the second computer readable program code;

wherein the fifth computer readable program code also stores the dictionary system quality data generated by the seventh computer readable program code in the identifier change database.

23. The medium of claim 17, wherein the first computer

readable program code stores a version of the dictionary data for which editing operations are in progress, into an editing dictionary database of the dictionary database; and

the computer readable program codes further includes an eighth computer readable program code for causing said computer to store a version of the dictionary data for which the editing operations are completed, into a master dictionary database of the dictionary database.

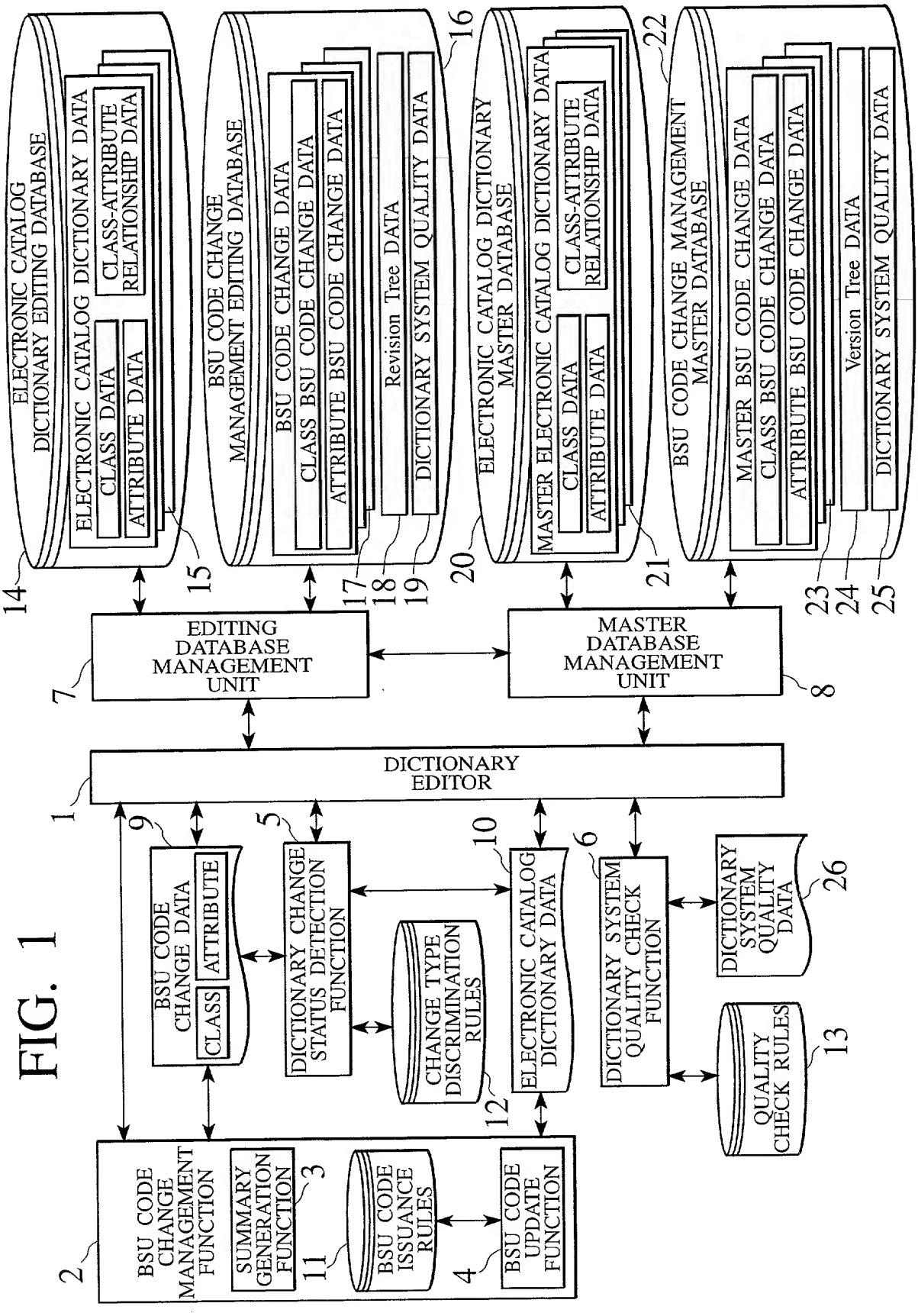
24. The medium of claim 17, wherein the first computer readable program code stores the identifier change data corresponding to a version of the dictionary data for which editing operations are in progress, into an editing identifier change database of the identifier change database; and

the computer readable program codes further includes an eighth computer readable program code for causing said computer to store the identifier change data corresponding to a version of the dictionary data for which the editing operations are completed, into a master identifier change database of the identifier change database.

ABSTRACT OF THE DISCLOSURE

In an electronic catalog maintenance system using the dictionary data given in a form of a tree structure formed by identifiers for uniquely identifying classes classifying products and attributes of the products, the dictionary data are edited by making changes including standard changes defined by a prescribed standard and out-of-standard changes not defined by the prescribed standard, and an identifier change data indicating the status of each change made by the editing and updates of identifiers to be made in the dictionary data is generated. Then, a new identifier of each class or attribute created by an out-of-standard change is issued and an old identifier of each class or attribute deleted by an out-of-standard change is retired according to the identifier change data, while storing the identifier change data in an identifier change database.

FIG. 1


ELECTRONIC CATALOG DICTIONARY EDITING DATABASE
ELECTRONIC CATALOG DICTIONARY DATA
CLASS DATA
ATTRIBUTE DATA
CLASS-ATTRIBUTE RELATIONSHIP DATA
BSU CODE CHANGE MANAGEMENT EDITING DATABASE
BSU CODE CHANGE DATA
CLASS BSU CODE CHANGE DATA
ATTRIBUTE BSU CODE CHANGE DATA
Revision Tree DATA
DICTIONARY SYSTEM QUALITY DATA
ELECTRONIC CATALOG DICTIONARY MASTER DATABASE
MASTER ELECTRONIC CATALOG DICTIONARY DATA
CLASS DATA
ATTRIBUTE DATA
CLASS-ATTRIBUTE RELATIONSHIP DATA
BSU CODE CHANGE MANAGEMENT MASTER DATABASE
MASTER BSU CODE CHANGE DATA
CLASS BSU CODE CHANGE DATA
ATTRIBUTE BSU CODE CHANGE DATA
Version Tree DATA
DICTIONARY SYSTEM QUALITY DATA
14
15
16
17
18
19
20
21
22
23
24
25
EDITING DATABASE MANAGEMENT UNIT
7
MASTER DATABASE MANAGEMENT UNIT
8
DICTIONARY EDITOR
1
BSU CODE CHANGE DATA
CLASS
ATTRIBUTE
9
DICTIONARY CHANGE STATUS DETECTION FUNCTION
5
CHANGE TYPE DISCRIMINATION RULES
12
ELECTRONIC CATALOG DICTIONARY DATA
10
DICTIONARY SYSTEM QUALITY CHECK FUNCTION
6
DICTIONARY SYSTEM QUALITY DATA
26
QUALITY CHECK RULES
13
BSU CODE CHANGE MANAGEMENT FUNCTION
2
SUMMARY GENERATION FUNCTION
3
BSU CODE ISSUANCE RULES
11
BSU CODE UPDATE FUNCTION
4

# FIG. 2

CLASS DATA

| CID | Preferred_ Name | BSU | PARENT CLASS CID | ..... | ..... | DATA QUALITY LEVEL |
|---|---|---|---|---|---|---|
| A0 | CLASS A0 | A001 | | | | A |
| B0 | CLASS B0 | A002 | A001 | | | A |
| B1 | CLASS B1 | A003 | A001 | | | A |
| C0 | CLASS C0 | A004 | A002 | | | B |
| C1 | CLASS C1 | A005 | A002 | | | B |
| C2 | CLASS C2 | A006 | A003 | | | C |
| C3 | CLASS C3 | A007 | A003 | | | D |

# FIG. 3

ATTRIBUTE DATA

| PID | Preferred_ Name | BSU | ..... | ..... | ..... | DATA QUALITY LEVEL |
|---|---|---|---|---|---|---|
| V0 | ATTRIBUTE V0 | P001 | | | | A |
| V1 | ATTRIBUTE V1 | P002 | | | | A |
| V2 | ATTRIBUTE V2 | P003 | | | | A |
| V3 | ATTRIBUTE V3 | P004 | | | | B |
| V4 | ATTRIBUTE V4 | P005 | | | | B |
| V5 | ATTRIBUTE V5 | P006 | | | | B |
| V6 | ATTRIBUTE V6 | P007 | | | | B |

# FIG. 4

CLASS-ATTRIBUTE RELATIONSHIP DATA

| PID | CID scope | Visible | Applicable |
|---|---|---|---|
| V0 | A0 | A0, B0, B1, C0, C1, C2, C3 | A0, B0, B1, C0, C1, C2, C3 |
| V1 | B0 | B0, C0, C1 | B0, C0, C1 |
| V2 | C0 | C0 | NULL |
| V3 | C1 | C1 | C1 |
| V4 | B1 | B1, C2, C3 | C3 |
| V5 | C2 | C2 | NULL |
| V6 | C3 | C3 | NULL |

# FIG. 5

CLASS BSU CODE CHANGE DATA

| Status | CID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | C4 | NULL | NULL | C3 |
| OOD | C3 | A007 | C4 | NULL |
| RUP | C2 | A006 | NULL | NULL |

※ VUP : Version UP
RUP : Revision UP
NEW : New BSU
OOD : Out of Dictionary

# FIG. 6

ATTRIBUTE BSU CODE CHANGE DATA

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| NEW | V8 | NULL | NULL | V6 |
| OOD | V6 | P007 | V8 | NULL |
| RUP | V2 | P003 | NULL | NULL |

FIG. 7

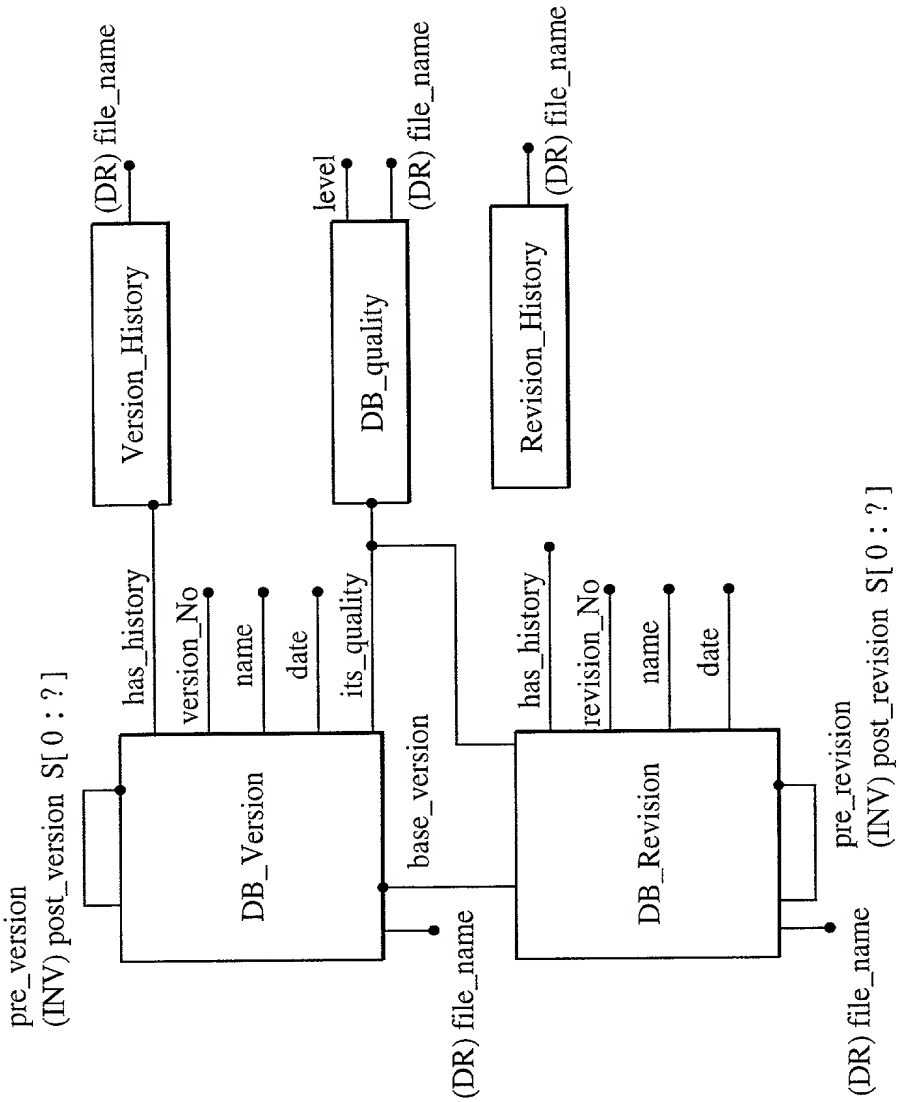
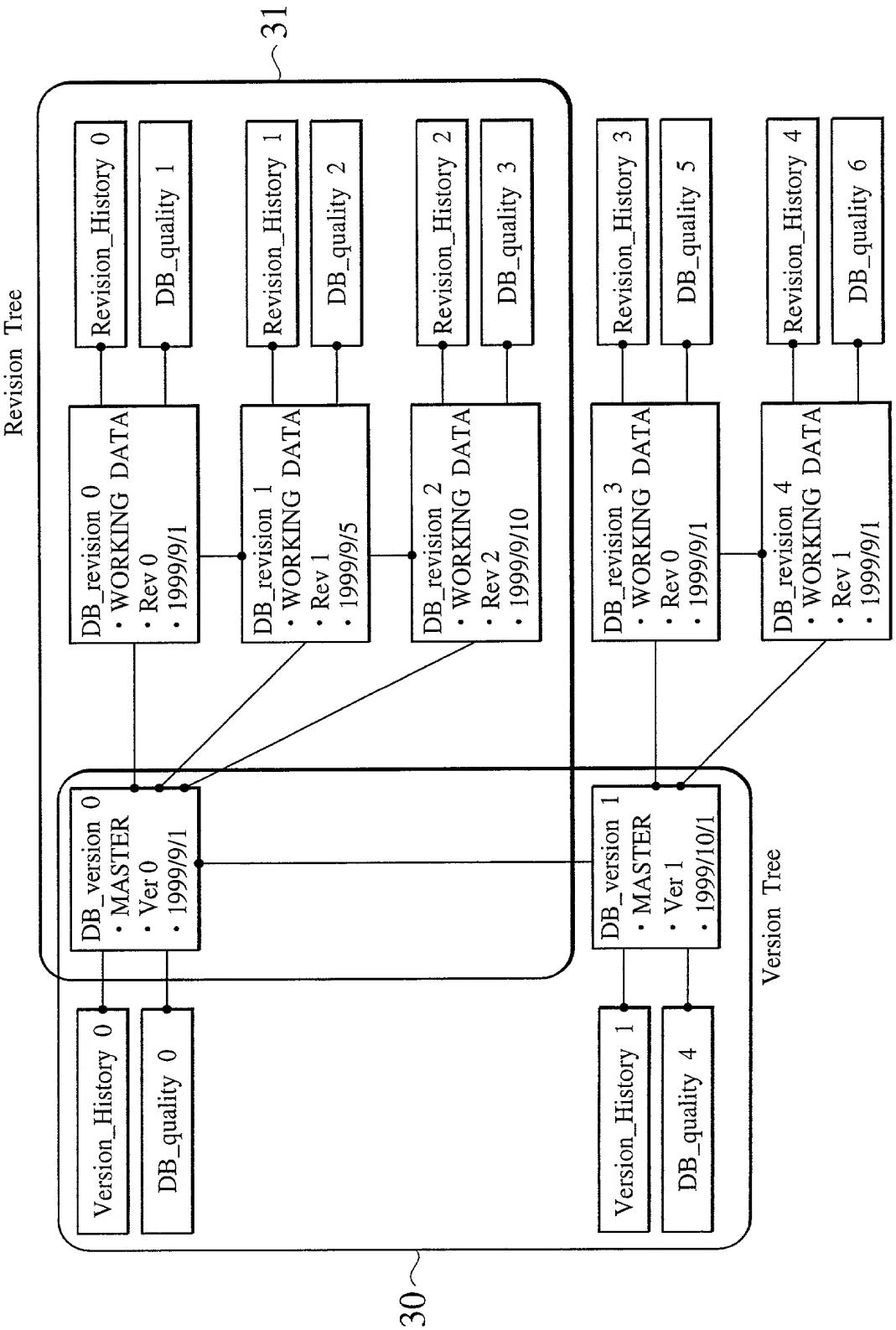

pre_version
(INV) post_version S[ 0 : ? ]
DB_Version
has_history
version_No
name
date
its_quality
(DR) file_name
Version_History
(DR) file_name
level
DB_quality
(DR) file_name
base_version
Revision_History
(DR) file_name
has_history
revision_No
name
date
DB_Revision
pre_revision
(INV) post_revision S[ 0 : ? ]
(DR) file_name

FIG. 8

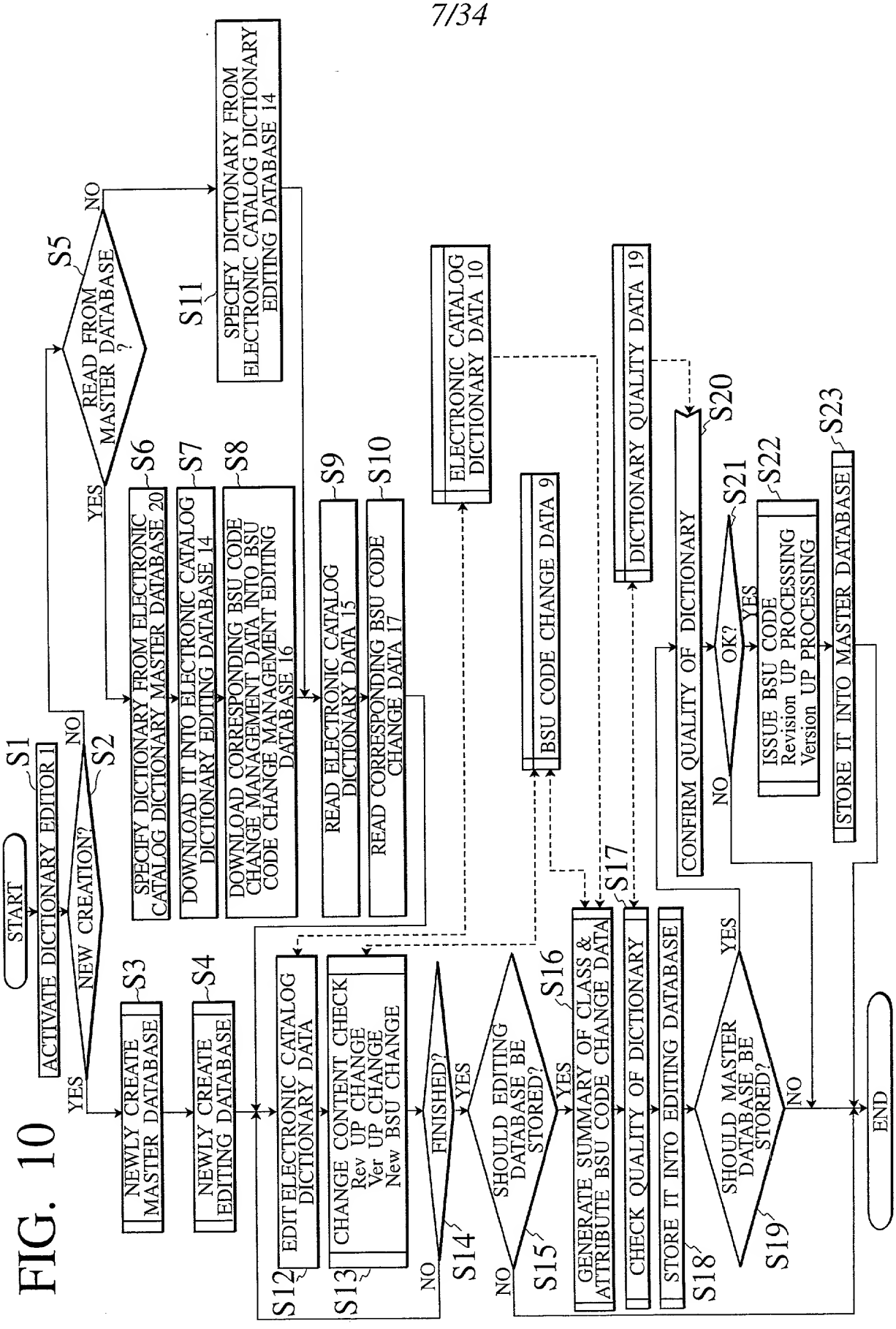

31
Revision Tree
Revision_History 0
DB_quality 1
Revision_History 1
DB_quality 2
Revision_History 2
DB_quality 3
Revision_History 3
DB_quality 5
Revision_History 4
DB_quality 6
DB_revision 0
• WORKING DATA
• Rev 0
• 1999/9/1
DB_revision 1
• WORKING DATA
• Rev 1
• 1999/9/5
DB_revision 2
• WORKING DATA
• Rev 2
• 1999/9/10
DB_revision 3
• WORKING DATA
• Rev 0
• 1999/9/1
DB_revision 4
• WORKING DATA
• Rev 1
• 1999/9/1
DB_version 0
• MASTER
• Ver 0
• 1999/9/1
DB_version 1
• MASTER
• Ver 1
• 1999/10/1
Version_History 0
DB_quality 0
Version_History 1
DB_quality 4
Version Tree
30

# FIG. 9

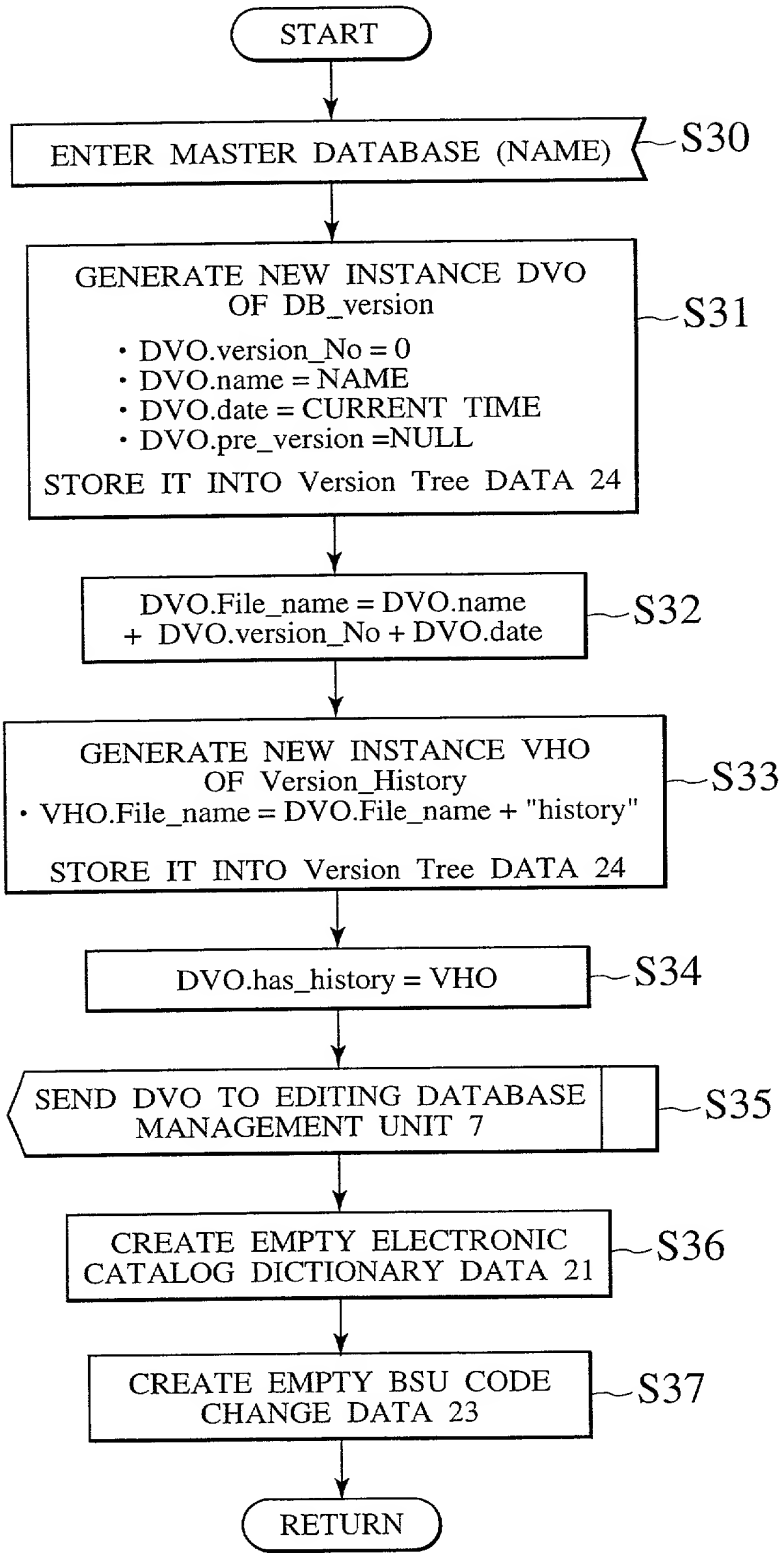
DICTIONARY SYSTEM QUALITY DATA

| ID | CONTENT |
|---|---|
| C1 | THERE IS ONE SUB-CLASS |
| V7 | CONTENT IS SAME AS V3 |
| V8 | CONTENT IS SAME AS V4 |
| V9 | CONTENT IS SAME AS V5 |
| CLASS SYSTEM | B |

FIG. 10


START
S1
ACTIVATE DICTIONARY EDITOR 1
S2
NEW CREATION?
YES
NO
S3
NEWLY CREATE MASTER DATABASE
S4
NEWLY CREATE EDITING DATABASE
S5
READ FROM MASTER DATABASE ?
YES
NO
S6
SPECIFY DICTIONARY FROM ELECTRONIC CATALOG DICTIONARY MASTER DATABASE 20
S7
DOWNLOAD IT INTO ELECTRONIC CATALOG DICTIONARY EDITING DATABASE 14
S8
DOWNLOAD CORRESPONDING BSU CODE CHANGE MANAGEMENT DATA INTO BSU CODE CHANGE MANAGEMENT EDITING DATABASE 16
S9
READ ELECTRONIC CATALOG DICTIONARY DATA 15
S10
READ CORRESPONDING BSU CODE CHANGE DATA 17
S11
SPECIFY DICTIONARY FROM ELECTRONIC CATALOG DICTIONARY EDITING DATABASE 14
S12
EDIT ELECTRONIC CATALOG DICTIONARY DATA
S13
CHANGE CONTENT CHECK
Rev UP CHANGE
Ver UP CHANGE
New BSU CHANGE
S14
FINISHED?
NO
YES
S15
SHOULD EDITING DATABASE BE STORED?
NO
YES
S16
GENERATE SUMMARY OF CLASS & ATTRIBUTE BSU CODE CHANGE DATA
S17
CHECK QUALITY OF DICTIONARY
S18
STORE IT INTO EDITING DATABASE
S19
SHOULD MASTER DATABASE BE STORED?
YES
NO
S20
CONFIRM QUALITY OF DICTIONARY
S21
OK?
NO
YES
S22
ISSUE BSU CODE
Revision UP PROCESSING
Version UP PROCESSING
S23
STORE IT INTO MASTER DATABASE
END
ELECTRONIC CATALOG DICTIONARY DATA 10
BSU CODE CHANGE DATA 9
DICTIONARY QUALITY DATA 19

# FIG. 11


START
ENTER MASTER DATABASE (NAME)
S30
GENERATE NEW INSTANCE DVO OF DB_version
• DVO.version_No = 0
• DVO.name = NAME
• DVO.date = CURRENT TIME
• DVO.pre_version =NULL
STORE IT INTO Version Tree DATA 24
S31
DVO.File_name = DVO.name + DVO.version_No + DVO.date
S32
GENERATE NEW INSTANCE VHO OF Version_History
• VHO.File_name = DVO.File_name + "history"
STORE IT INTO Version Tree DATA 24
S33
DVO.has_history = VHO
S34
SEND DVO TO EDITING DATABASE MANAGEMENT UNIT 7
S35
CREATE EMPTY ELECTRONIC CATALOG DICTIONARY DATA 21
S36
CREATE EMPTY BSU CODE CHANGE DATA 23
S37
RETURN

# FIG. 12

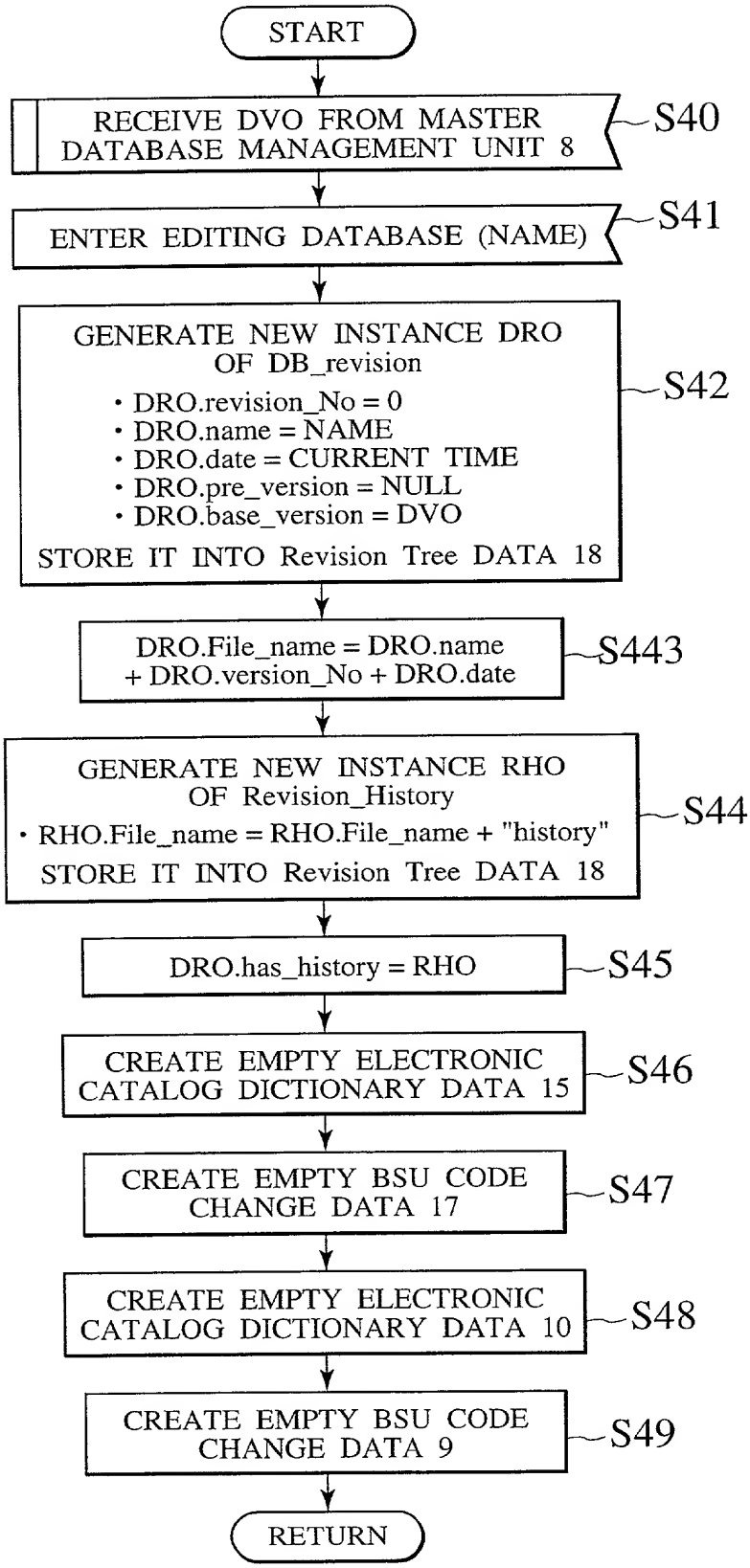
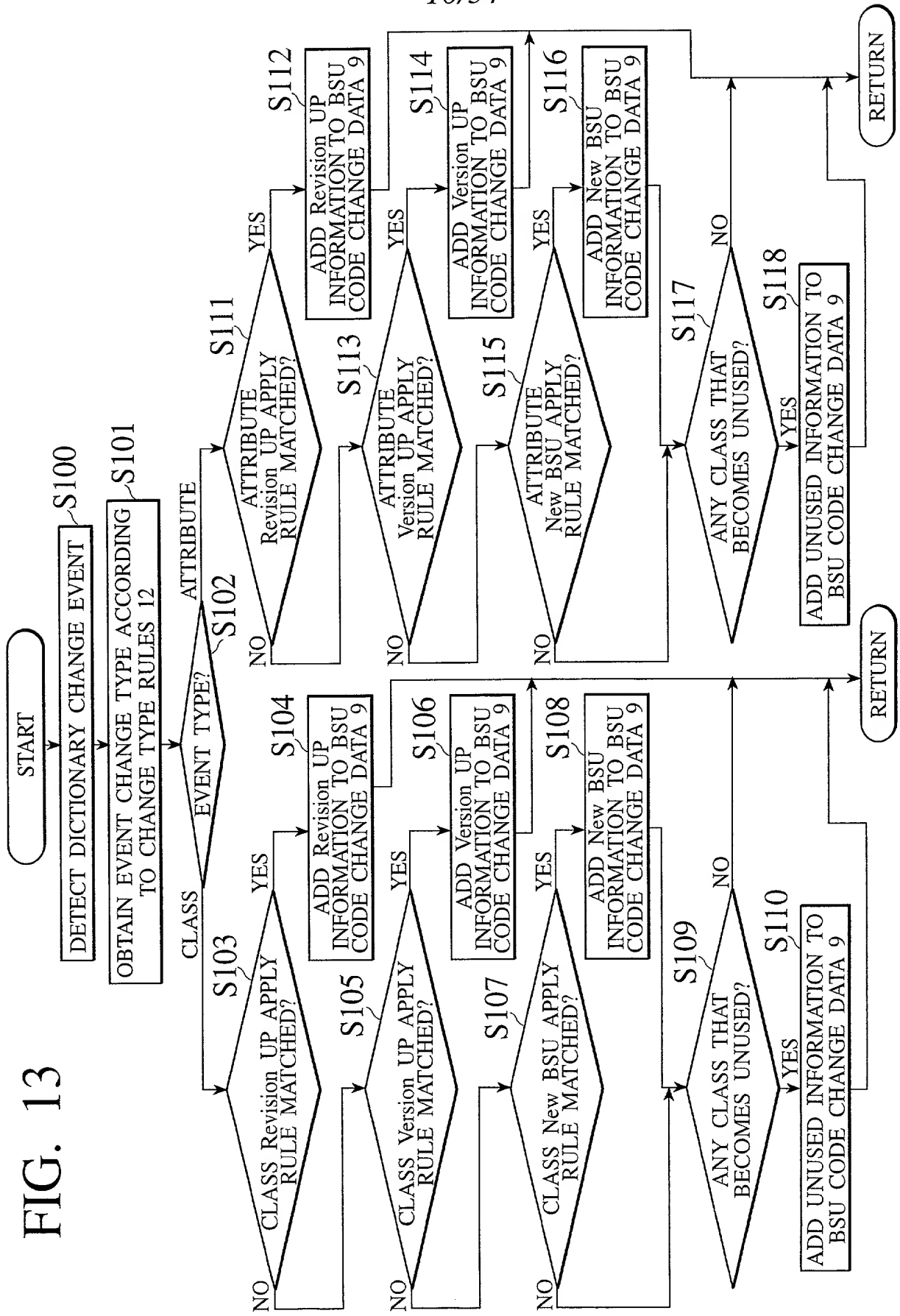

START
RECEIVE DVO FROM MASTER DATABASE MANAGEMENT UNIT 8
S40
ENTER EDITING DATABASE (NAME)
S41
GENERATE NEW INSTANCE DRO OF DB_revision
• DRO.revision_No = 0
• DRO.name = NAME
• DRO.date = CURRENT TIME
• DRO.pre_version = NULL
• DRO.base_version = DVO
STORE IT INTO Revision Tree DATA 18
S42
DRO.File_name = DRO.name + DRO.version_No + DRO.date
S443
GENERATE NEW INSTANCE RHO OF Revision_History
• RHO.File_name = RHO.File_name + "history"
STORE IT INTO Revision Tree DATA 18
S44
DRO.has_history = RHO
S45
CREATE EMPTY ELECTRONIC CATALOG DICTIONARY DATA 15
S46
CREATE EMPTY BSU CODE CHANGE DATA 17
S47
CREATE EMPTY ELECTRONIC CATALOG DICTIONARY DATA 10
S48
CREATE EMPTY BSU CODE CHANGE DATA 9
S49
RETURN

FIG. 13


START
DETECT DICTIONARY CHANGE EVENT
S100
OBTAIN EVENT CHANGE TYPE ACCORDING TO CHANGE TYPE RULES 12
S101
ATTRIBUTE
CLASS
EVENT TYPE?
S102
CLASS Revision UP APPLY RULE MATCHED?
S103
YES
NO
ADD Revision UP INFORMATION TO BSU CODE CHANGE DATA 9
S104
CLASS Version UP APPLY RULE MATCHED?
S105
ADD Version UP INFORMATION TO BSU CODE CHANGE DATA 9
S106
CLASS New BSU APPLY RULE MATCHED?
S107
ADD New BSU INFORMATION TO BSU CODE CHANGE DATA 9
S108
ANY CLASS THAT BECOMES UNUSED?
S109
ADD UNUSED INFORMATION TO BSU CODE CHANGE DATA 9
S110
RETURN
ATTRIBUTE
Revision UP APPLY RULE MATCHED?
S111
ADD Revision UP INFORMATION TO BSU CODE CHANGE DATA 9
S112
ATTRIBUTE
Version UP APPLY RULE MATCHED?
S113
ADD Version UP INFORMATION TO BSU CODE CHANGE DATA 9
S114
ATTRIBUTE
New BSU APPLY RULE MATCHED?
S115
ADD New BSU INFORMATION TO BSU CODE CHANGE DATA 9
S116
ANY CLASS THAT BECOMES UNUSED?
S117
ADD UNUSED INFORMATION TO BSU CODE CHANGE DATA 9
S118
RETURN

# FIG. 14

| | |
|---|---|
| RULE 1 : | IF (CHANGE OF PRODUCT CLASS) AND (CHANGE THAT IS Revision UP AS DESCRIBED IN ISO 13584-42) THEN (Revision UP OF PRODUCT CLASS) |
| RULE 2 : | IF (CHANGE OF PRODUCT CLASS) AND (CHANGE THAT IS Version UP AS DESCRIBED IN ISO 13584-42) THEN (Version UP OF PRODUCT CLASS) |
| RULE 3 : | IF (NEW CREATION OF END PRODUCT CLASS) THEN (New BSU OF OF PRODUCT CLASS) |
| RULE 4 : | IF (MERGE OF PRODUCT CLASS) THEN (New BSU OF PRODUCT CLASS) AND (UNUSED CLASS = PRODUCT CLASS TO BE MERGED) |
| RULE 5 : | IF (MOVE OF PRODUCT CLASS) AND (TOPOLOGY CHANGE) THEN (New BSU OF PRODUCT CLASS) AND (UNUSED CLASS = PRODUCT CLASS BEFORE MOVING) |
| RULE 6 : | IF (DELETE OF INTERMEDIATE PRODUCT CLASS) THEN (New BSU OF PRODUCT CLASS) AND (UNUSED CLASS = PRODUCT CLASS TO BE DELETED) |
| RULE 7 : | IF (INSERT OF INTERMEDIATE PRODUCT CLASS) THEN (New BSU OF PRODUCT CLASS) |
| RULE 8 : | IF (DELETE OF END PRODUCT CLASS) THEN (UNUSED CLASS = PRODUCT CLASS TO BE DELETED) |
| RULE 9 : | IF (CHANGE OF ATTRIBUTE) AND (CHANGE THAT IS Revision UP AS DESCRIBED IN ISO 13584-42) THEN (Revision UP OF ATTRIBUTE) |
| RULE 10 : | IF (CHANGE OF ATTRIBUTE) AND (CHANGE THAT IS Version UP AS DESCRIBED IN ISO 13584-42) THEN (Version UP OF ATTRIBUTE) |
| RULE 11 : | IF (NEW CREATION OF ATTRIBUTE) THEN (New BSU OF ATTRIBUTE) |
| RULE 12 : | IF (DELETE OF ATTRIBUTE) THEN (UNUSED ATTRIBUTE = ATTRIBUTE TO BE DELETED) |
| RULE 13 : | IF (CHANGE OF Name Scope OF ATTRIBUTE) THEN (New BSU OF ATTRIBUTE) AND (UNUSED ATTRIBUTE = ATTRIBUTE BEFORE CHANGE) |

FIG. 15

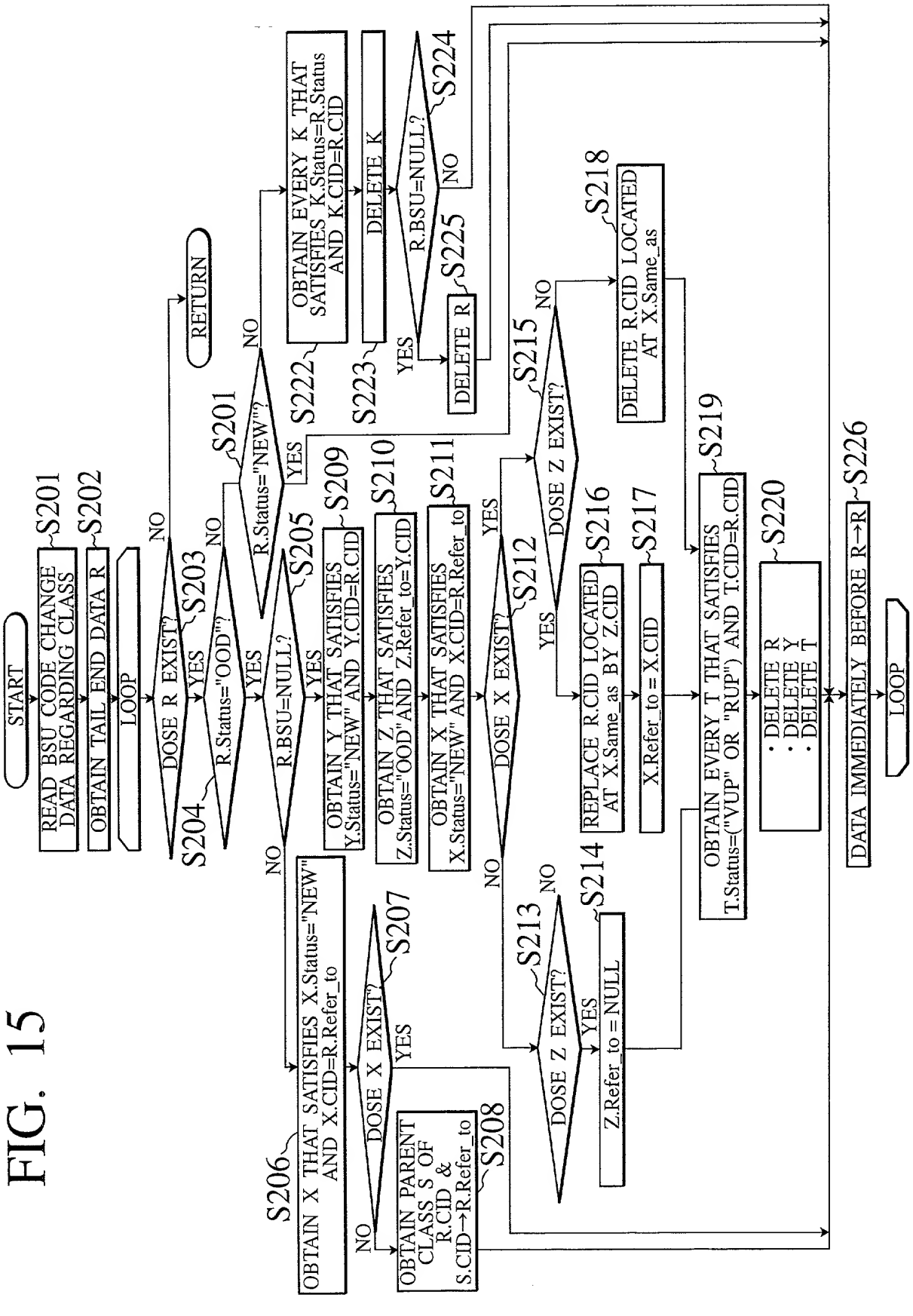
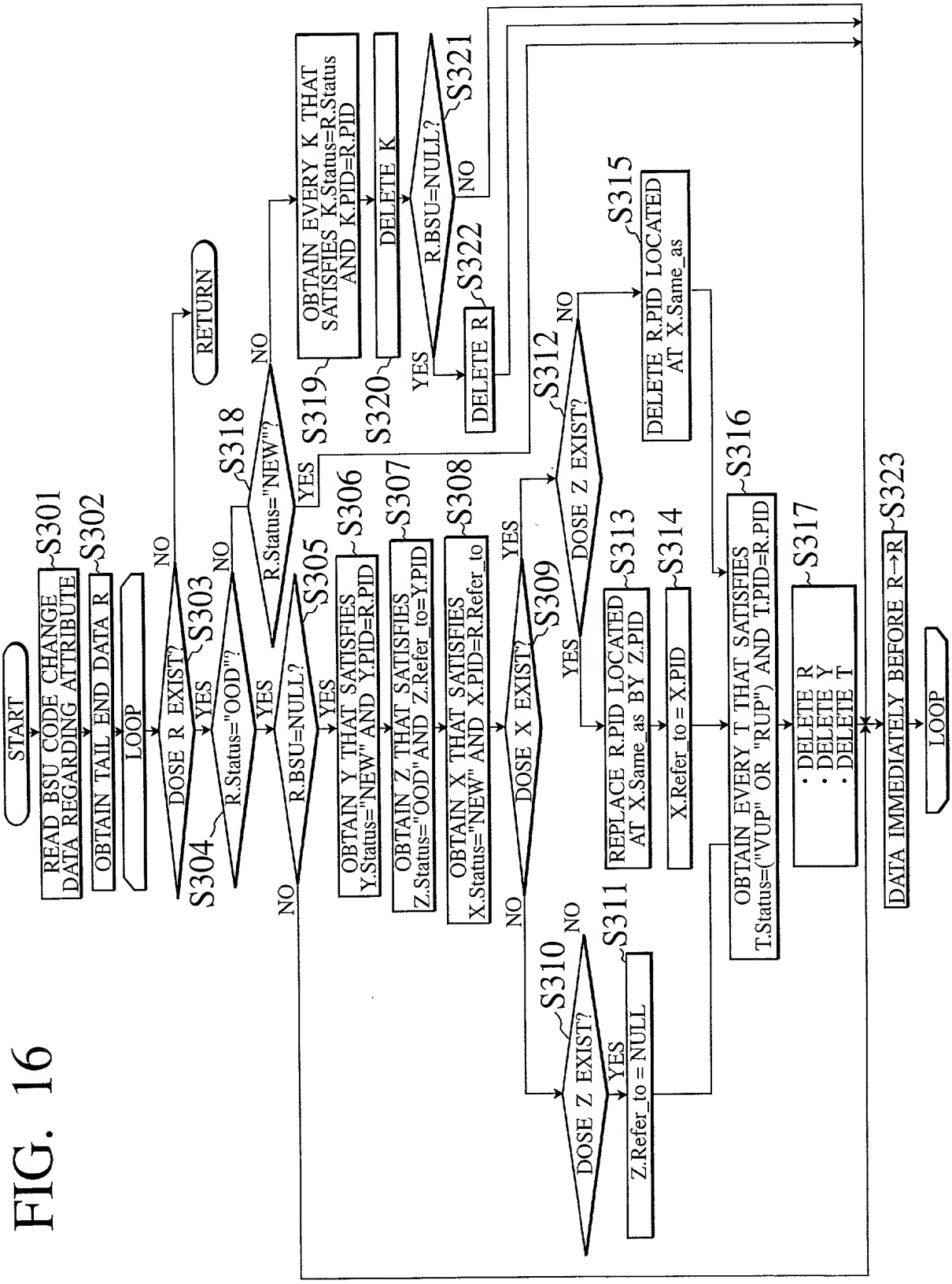
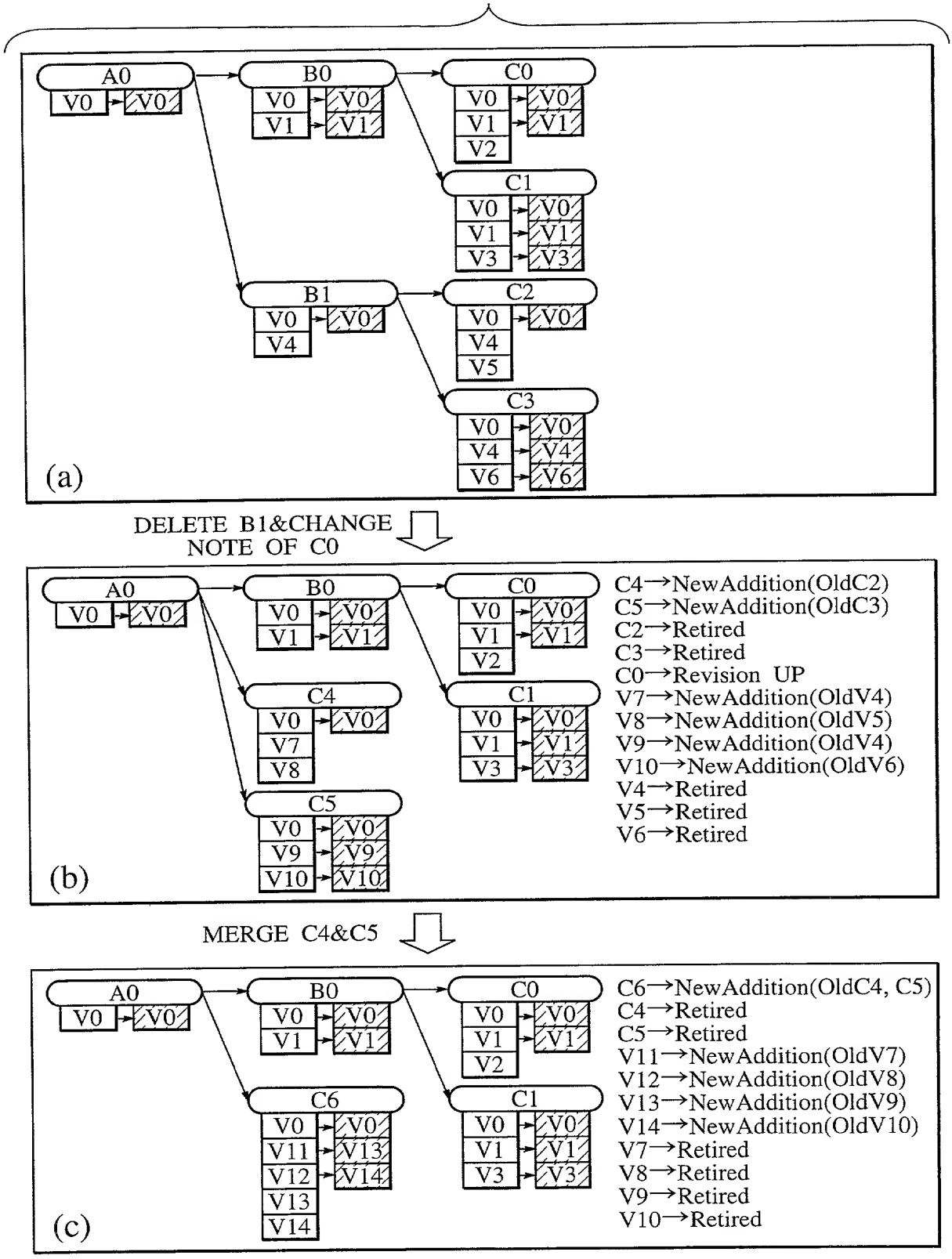

START
READ BSU CODE CHANGE DATA REGARDING CLASS
S201
OBTAIN TAIL END DATA R
S202
LOOP
DOSE R EXIST?
S203
NO
YES
RETURN
R.Status="OOD"?
S204
NO
YES
R.Status="NEW"?
S201
NO
YES
R.BSU=NULL?
S205
NO
YES
OBTAIN Y THAT SATISFIES Y.Status="NEW" AND Y.CID=R.CID
S209
OBTAIN Z THAT SATISFIES Z.Status="OOD" AND Z.Refer_to=Y.CID
S210
OBTAIN X THAT SATISFIES X.Status="NEW" AND X.CID=R.Refer_to
S211
OBTAIN X THAT SATISFIES X.Status="NEW" AND X.CID=R.CID
S206
DOSE X EXIST?
S207
NO
YES
OBTAIN PARENT CLASS S OF R.CID & S.CID→R.Refer_to
S208
DOSE X EXIST?
S212
NO
YES
DOSE Z EXIST?
S213
NO
YES
Z.Refer_to = NULL
S214
REPLACE R.CID LOCATED AT X.Same_as BY Z.CID
S216
X.Refer_to = X.CID
S217
OBTAIN EVERY T THAT SATISFIES T.Status=("VUP" OR "RUP") AND T.CID=R.CID
S219
· DELETE R
· DELETE Y
· DELETE T
S220
DOSE Z EXIST?
S215
NO
YES
DELETE R.CID LOCATED AT X.Same_as
S218
DATA IMMEDIATELY BEFORE R→R
S226
LOOP
OBTAIN EVERY K THAT SATISFIES K.Status=R.Status AND K.CID=R.CID
S222
DELETE K
S223
R.BSU=NULL?
S224
NO
YES
DELETE R
S225

FIG. 16


START
READ BSU CODE CHANGE DATA REGARDING ATTRIBUTE
S301
OBTAIN TAIL END DATA R
S302
LOOP
DOSE R EXIST?
S303
NO
YES
RETURN
R.Status="OOD"?
S304
NO
YES
R.BSU=NULL?
S305
NO
YES
OBTAIN Y THAT SATISFIES Y.Status="NEW" AND Y.PID=R.PID
S306
OBTAIN Z THAT SATISFIES Z.Status="OOD"AND Z.Refer_to=Y.PID
S307
OBTAIN X THAT SATISFIES X.Status="NEW" AND X.PID=R.Refer_to
S308
DOSE X EXIST?
S309
NO
YES
DOSE Z EXIST?
S310
NO
YES
Z.Refer_to = NULL
S311
REPLACE R.PID LOCATED AT X.Same_as BY Z.PID
S313
X.Refer_to = X.PID
S314
DOSE Z EXIST?
S312
NO
YES
DELETE R.PID LOCATED AT X.Same_as
S315
OBTAIN EVERY T THAT SATISFIES T.Status=("VUP" OR "RUP") AND T.PID=R.PID
S316
: DELETE R
: DELETE Y
: DELETE T
S317
R.Status="NEW"?
S318
NO
YES
OBTAIN EVERY K THAT SATISFIES K.Status=R.Status AND K.PID=R.PID
S319
DELETE K
S320
R.BSU=NULL?
S321
NO
YES
DELETE R
S322
DATA IMMEDIATELY BEFORE R→R
S323
LOOP

# FIG. 17


A0
V0
V0
B0
V0
V1
V0
V1
C0
V0
V1
V2
V0
V1
C1
V0
V1
V3
V0
V1
V3
B1
V0
V4
V0
C2
V0
V4
V5
V0
C3
V0
V4
V6
V0
V4
V6
(a)
DELETE B1&CHANGE
NOTE OF C0
C4→NewAddition(OldC2)
C5→NewAddition(OldC3)
C2→Retired
C3→Retired
C0→Revision UP
V7→NewAddition(OldV4)
V8→NewAddition(OldV5)
V9→NewAddition(OldV4)
V10→NewAddition(OldV6)
V4→Retired
V5→Retired
V6→Retired
(b)
MERGE C4&C5
C6→NewAddition(OldC4, C5)
C4→Retired
C5→Retired
V11→NewAddition(OldV7)
V12→NewAddition(OldV8)
V13→NewAddition(OldV9)
V14→NewAddition(OldV10)
V7→Retired
V8→Retired
V9→Retired
V10→Retired
(c)

# FIG. 18

CLASS BSU CODE CHANGE DATA

(a)

| Status | CID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | C4 | NULL | NULL | C2 |
| NEW | C5 | NULL | NULL | C3 |
| OOD | C2 | A007 | C4 | NULL |
| OOD | C3 | A008 | C5 | NULL |
| RUP | C0 | A006 | NULL | NULL |
| NEW | C6 | NULL | NULL | (C4, C5) |
| OOD | C4 | NULL | C6 | NULL |
| OOD | C5 | NULL | C6 | NULL |

⇩

(b)

| Status | CID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | C4 | NULL | NULL | C2 |
| NEW | C5 | NULL | NULL | C3 |
| OOD | C2 | A007 | C4 | NULL |
| OOD | C3 | A008 | C6 | NULL |
| RUP | C0 | A006 | NULL | NULL |
| NEW | C6 | NULL | NULL | (C4, C3) |
| OOD | C4 | NULL | C6 | NULL |
| OOD | C5 | NULL | C6 | NULL |

⇩

(c)

| Status | CID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | C4 | NULL | NULL | C2 |
| OOD | C2 | A007 | C4 | NULL |
| OOD | C3 | A008 | C6 | NULL |
| RUP | C0 | A006 | NULL | NULL |
| NEW | C6 | NULL | NULL | (C4, C3) |
| OOD | C4 | NULL | C6 | NULL |

⇩

(d)

| Status | CID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | C4 | NULL | NULL | C2 |
| OOD | C2 | A007 | C6 | NULL |
| OOD | C3 | A008 | C6 | NULL |
| RUP | C0 | A006 | NULL | NULL |
| NEW | C6 | NULL | NULL | (C2, C3) |
| OOD | C4 | NULL | C6 | NULL |

⇩

(e)

| Status | CID | BSU | Refer to | Same as |
|---|---|---|---|---|
| OOD | C2 | A007 | C6 | NULL |
| OOD | C3 | A008 | C6 | NULL |
| RUP | C0 | A006 | NULL | NULL |
| NEW | C6 | NULL | NULL | (C2, C3) |

# FIG. 19

ATTRIBUTE BSU CODE CHANGE DATA

(a)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| NEW | V8 | NULL | NULL | V5 |
| NEW | V9 | NULL | NULL | V4 |
| NEW | V10 | NULL | NULL | V6 |
| OOD | V4 | P004 | (V7, V9) | NULL |
| OOD | V5 | P005 | V8 | NULL |
| OOD | V6 | P006 | V10 | NULL |
| NEW | V11 | NULL | NULL | V4 |
| NEW | V12 | NULL | NULL | V5 |
| NEW | V13 | NULL | NULL | V4 |
| NEW | V14 | NULL | NULL | V6 |
| OOD | V7 | NULL | V11 | NULL |
| OOD | V8 | NULL | V12 | NULL |
| OOD | V9 | NULL | V13 | NULL |
| OOD | V10 | NULL | V14 | NULL |

(b)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| NEW | V8 | NULL | NULL | V5 |
| NEW | V9 | NULL | NULL | V4 |
| NEW | V10 | NULL | NULL | V6 |
| OOD | V4 | P004 | (V7, V9) | NULL |
| OOD | V5 | P005 | V8 | NULL |
| OOD | V6 | P006 | V14 | NULL |
| NEW | V11 | NULL | NULL | V7 |
| NEW | V12 | NULL | NULL | V8 |
| NEW | V13 | NULL | NULL | V9 |
| NEW | V14 | NULL | NULL | V6 |
| OOD | V7 | NULL | V11 | NULL |
| OOD | V8 | NULL | V12 | NULL |
| OOD | V9 | NULL | V13 | NULL |
| OOD | V10 | NULL | V14 | NULL |

(c)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| NEW | V8 | NULL | NULL | V5 |
| NEW | V9 | NULL | NULL | V4 |
| OOD | V4 | P004 | (V7, V9) | NULL |
| OOD | V5 | P005 | V8 | NULL |
| OOD | V6 | P006 | V14 | NULL |
| NEW | V11 | NULL | NULL | V7 |
| NEW | V12 | NULL | NULL | V8 |
| NEW | V13 | NULL | NULL | V9 |
| NEW | V14 | NULL | NULL | V6 |
| OOD | V7 | NULL | V11 | NULL |
| OOD | V8 | NULL | V12 | NULL |
| OOD | V9 | NULL | V13 | NULL |

(d)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| NEW | V8 | NULL | NULL | V5 |
| NEW | V9 | NULL | NULL | V4 |
| OOD | V4 | P004 | (V7, V13) | NULL |
| OOD | V5 | P005 | V8 | NULL |
| OOD | V6 | P006 | V14 | NULL |
| NEW | V11 | NULL | NULL | V7 |
| NEW | V12 | NULL | NULL | V8 |
| NEW | V13 | NULL | NULL | V4 |
| NEW | V14 | NULL | NULL | V6 |
| OOD | V7 | NULL | V11 | NULL |
| OOD | V8 | NULL | V12 | NULL |
| OOD | V9 | NULL | V13 | NULL |

(e)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| NEW | V8 | NULL | NULL | V5 |
| OOD | V4 | P004 | (V7, V13) | NULL |
| OOD | V5 | P005 | V8 | NULL |
| OOD | V6 | P006 | V14 | NULL |
| NEW | V11 | NULL | NULL | V7 |
| NEW | V12 | NULL | NULL | V8 |
| NEW | V13 | NULL | NULL | V4 |
| NEW | V14 | NULL | NULL | V6 |
| OOD | V7 | NULL | V11 | NULL |
| OOD | V8 | NULL | V12 | NULL |

(f)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| NEW | V8 | NULL | NULL | V5 |
| OOD | V4 | P004 | (V7, V13) | NULL |
| OOD | V5 | P005 | V12 | NULL |
| OOD | V6 | P006 | V14 | NULL |
| NEW | V11 | NULL | NULL | V7 |
| NEW | V12 | NULL | NULL | V5 |
| NEW | V13 | NULL | NULL | V4 |
| NEW | V14 | NULL | NULL | V6 |
| OOD | V7 | NULL | V11 | NULL |
| OOD | V8 | NULL | V12 | NULL |

(g)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| OOD | V4 | P004 | (V7, V13) | NULL |
| OOD | V5 | P005 | V12 | NULL |
| OOD | V6 | P006 | V14 | NULL |
| NEW | V11 | NULL | NULL | V7 |
| NEW | V12 | NULL | NULL | V5 |
| NEW | V13 | NULL | NULL | V4 |
| NEW | V14 | NULL | NULL | V6 |
| OOD | V7 | NULL | V11 | NULL |

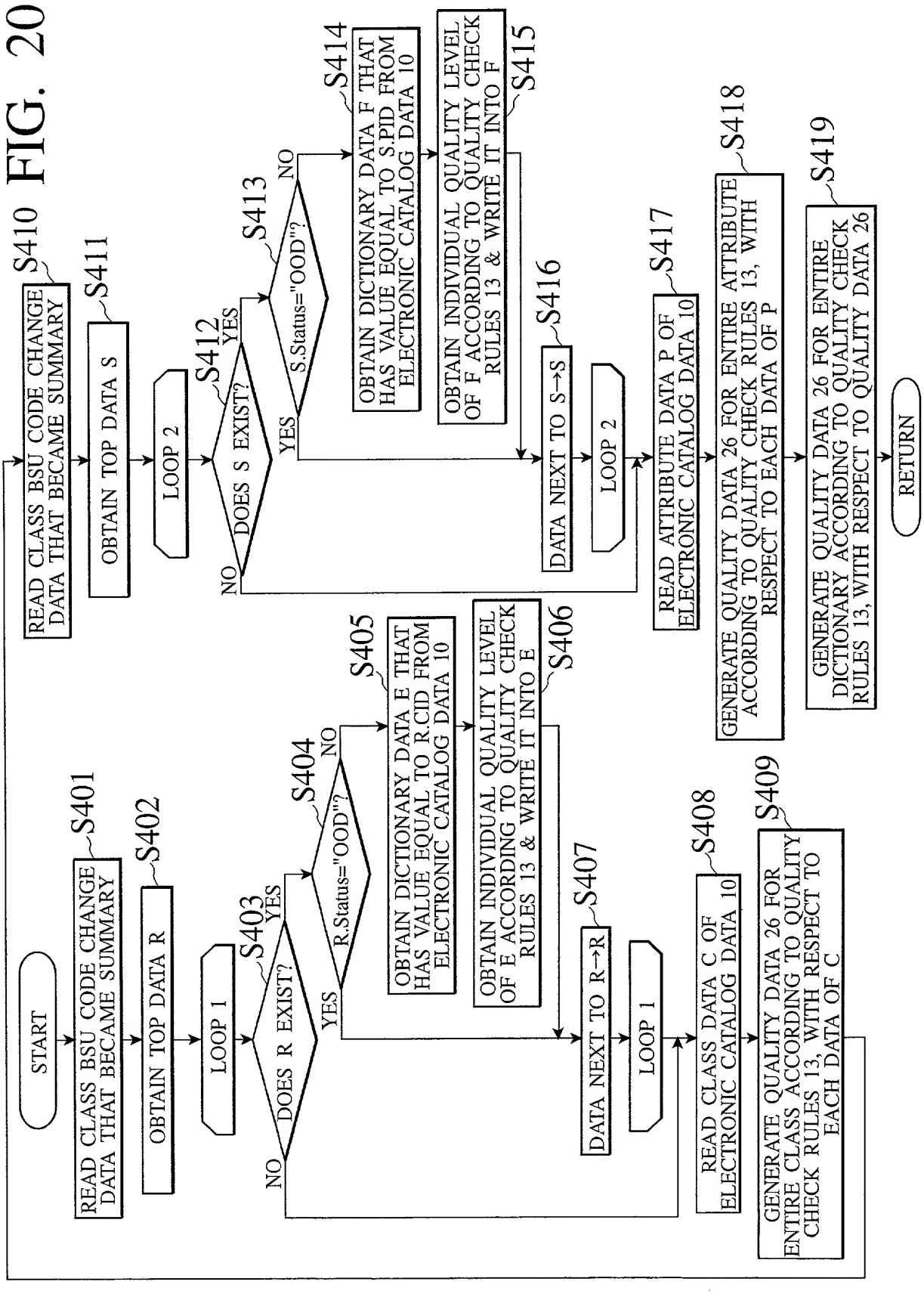
(h)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| NEW | V7 | NULL | NULL | V4 |
| OOD | V4 | P004 | (V7, V13) | NULL |
| OOD | V5 | P005 | V12 | NULL |
| OOD | V6 | P006 | V14 | NULL |
| NEW | V11 | NULL | NULL | V4 |
| NEW | V12 | NULL | NULL | V5 |
| NEW | V13 | NULL | NULL | V4 |
| NEW | V14 | NULL | NULL | V6 |
| OOD | V7 | NULL | V11 | NULL |

(i)

| Status | PID | BSU | Refer to | Same as |
|---|---|---|---|---|
| OOD | V4 | P004 | (V11, V13) | NULL |
| OOD | V5 | P005 | V12 | NULL |
| OOD | V6 | P006 | V14 | NULL |
| NEW | V11 | NULL | NULL | V4 |
| NEW | V12 | NULL | NULL | V5 |
| NEW | V13 | NULL | NULL | V4 |
| NEW | V14 | NULL | NULL | V6 |

FIG. 20


START
READ CLASS BSU CODE CHANGE DATA THAT BECAME SUMMARY
S401
OBTAIN TOP DATA R
S402
LOOP 1
S403
DOES R EXIST?
YES
NO
R.Status="OOD"?
S404
YES
NO
OBTAIN DICTIONARY DATA E THAT HAS VALUE EQUAL TO R.CID FROM ELECTRONIC CATALOG DATA 10
S405
OBTAIN INDIVIDUAL QUALITY LEVEL OF E ACCORDING TO QUALITY CHECK RULES 13 & WRITE IT INTO E
S406
DATA NEXT TO R→R
S407
LOOP 1
READ CLASS DATA C OF ELECTRONIC CATALOG DATA 10
S408
GENERATE QUALITY DATA 26 FOR ENTIRE CLASS ACCORDING TO QUALITY CHECK RULES 13, WITH RESPECT TO EACH DATA OF C
S409
READ CLASS BSU CODE CHANGE DATA THAT BECAME SUMMARY
S410
OBTAIN TOP DATA S
S411
LOOP 2
S412
DOES S EXIST?
YES
NO
S.Status="OOD"?
S413
YES
NO
OBTAIN DICTIONARY DATA F THAT HAS VALUE EQUAL TO S.PID FROM ELECTRONIC CATALOG DATA 10
S414
OBTAIN INDIVIDUAL QUALITY LEVEL OF F ACCORDING TO QUALITY CHECK RULES 13 & WRITE IT INTO F
S415
DATA NEXT TO S→S
S416
LOOP 2
READ ATTRIBUTE DATA P OF ELECTRONIC CATALOG DATA 10
S417
GENERATE QUALITY DATA 26 FOR ENTIRE ATTRIBUTE ACCORDING TO QUALITY CHECK RULES 13, WITH RESPECT TO EACH DATA OF P
S418
GENERATE QUALITY DATA 26 FOR ENTIRE DICTIONARY ACCORDING TO QUALITY CHECK RULES 13, WITH RESPECT TO QUALITY DATA 26
S419
RETURN

# FIG. 21A

| | |
|---|---|
| RULE 1 : | IF (E=PRODUCT CLASS) AND (INDIVIDUAL CHECK)<br>THEN(E.QUALITY LEVEL=EVALUATION FUNCTION F1 (E)) |
| RULE 2 : | IF (E=ATTRIBUTE CLASS) AND (INDIVIDUAL CHECK)<br>THEN(E.QUALITY LEVEL=EVALUATION FUNCTION F2(E)) |
| RULE 3 : | IF (E=PRODUCT CLASS) AND (SYSTEM CHECK)<br>THEN(SYSTEM QUALITY LEVEL L1=EVALUATION FUNCTION F3 (E)) |
| RULE 4 : | IF (E=ATTRIBUTE CLASS) AND (SYSTEM CHECK)<br>THEN(SYSTEM QUALITY LEVEL L1=EVALUATION FUNCTION F4(E)) |
| RULE 5: | IF (E=DICTIONARY)<br>THEN(SYSTEM QUALITY LEVEL L1=EVALUATION FUNCTION F5(E)) |

| | |
|---|---|
| EVALUATION FUNCTION 1: | IF((Y=# OF ATTRIBUTES DESCRIBED IN X/ # OF ATTRIBUTES OF X)>0.9)THEN QUALITY LEVEL="A" |
| | ELSE IF (0.9≧Y>0.6) THEN QUALITY LEVEL="B" |
| | ELSE IF (0.6≧Y>0.4) THEN QUALITY LEVEL="C" |
| | ELSE IF (0.4≧Y) THEN QUALITY LEVEL="D" |
| EVALUATION FUNCTION 2: | IF((Y=# OF ATTRIBUTES DESCRIBED IN X/ # OF ATTRIBUTES OF X)>0.95)THEN QUALITY LEVEL="A" |
| | ELSE IF (0.95≧Y>0.8) THEN QUALITY LEVEL="B" |
| | ELSE IF (0.8≧Y>0.6) THEN QUALITY LEVEL="C" |
| | ELSE IF (0.6≧Y) THEN QUALITY LEVEL="D" |
| EVALUATION FUNCTION 3: | IF (X!= END CLASS) AND (# OF SUB-CLASSES OF X=1) |
| | THEN L1=L1+1, Return ("X.CID", "THERE IS ONE SUB_CLASS") |
| EVALUATION FUNCTION 4: | IF (X.Same-as!=NULL) AND<br>(NOT (Z.CID=X.Same_as) AND (Z.Status="OOD")) |
| | THEN L1=L1+1, Return ("X.CID", "SAME AS X.Same_as")) |
| EVALUATION FUNCTION 5: | IF (L1<5) THEN QUALITY LEVEL = "A" |
| | ELSE IF (10≧L1>5) THEN QUALITY LEVEL="B" |
| | ELSE IF (50≧L1>10) THEN QUALITY LEVEL="C" |
| | ELSE IF (50>L1) THEN QUALITY LEVEL="D" |

# FIG. 21B

| | |
|---|---|
| RULE 1 : | IF ISSUING OF CLASS BSU<br>THEN BSU CODE ="A"+MAX OF CURRENT CLASS BSU ISSUED No.+1 |
| RULE 2 : | IF ISSUING OF ATTRIBUTE BSU<br>THEN BSU CODE = "P"+MAX OF CURRENT ATTRIBUTE BSU ISSUED No.+1 |

FIG. 22

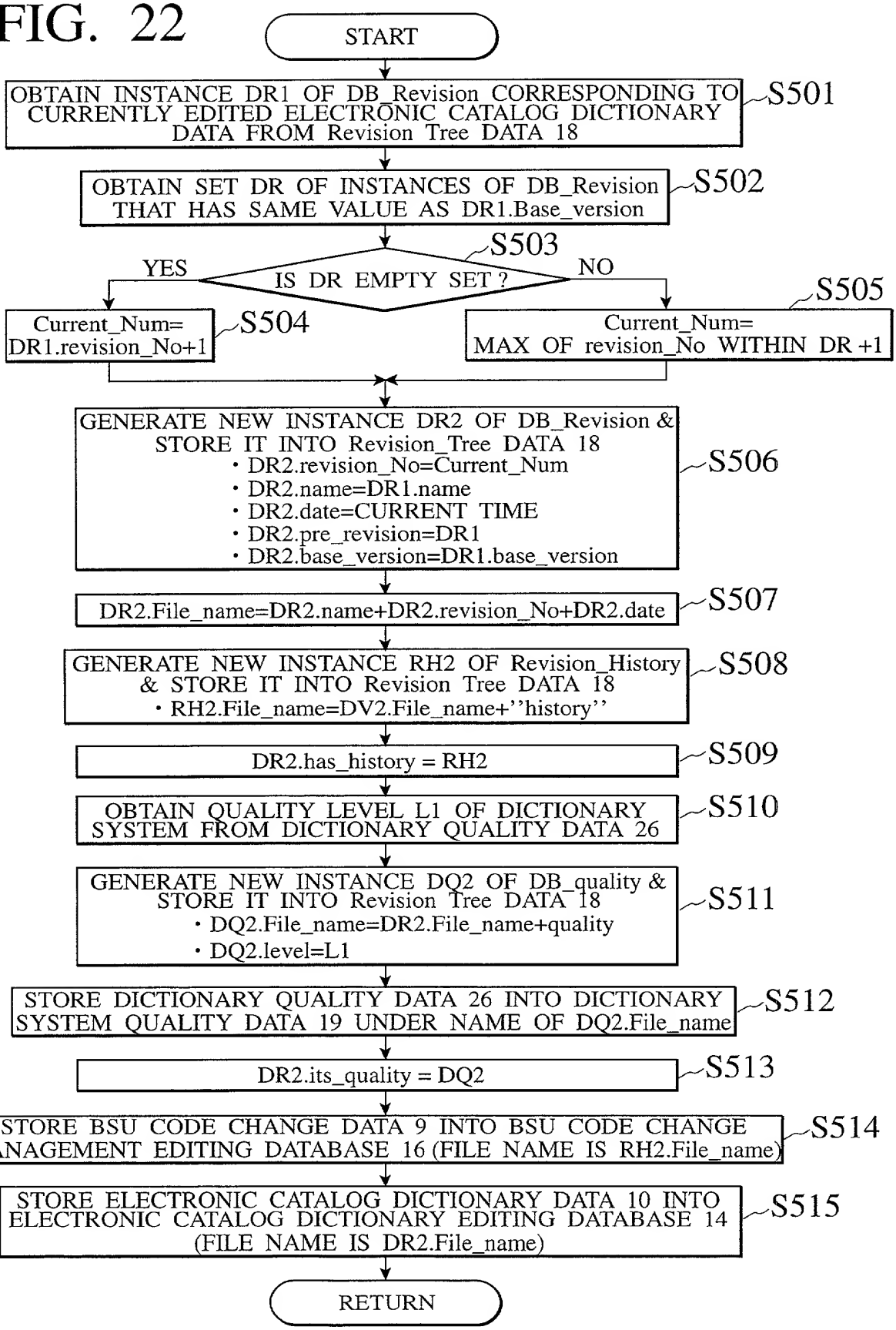

START
OBTAIN INSTANCE DR1 OF DB_Revision CORRESPONDING TO CURRENTLY EDITED ELECTRONIC CATALOG DICTIONARY DATA FROM Revision Tree DATA 18
S501
OBTAIN SET DR OF INSTANCES OF DB_Revision THAT HAS SAME VALUE AS DR1.Base_version
S502
S503
IS DR EMPTY SET ?
YES
NO
Current_Num= DR1.revision_No+1
S504
Current_Num= MAX OF revision_No WITHIN DR +1
S505
GENERATE NEW INSTANCE DR2 OF DB_Revision & STORE IT INTO Revision_Tree DATA 18
• DR2.revision_No=Current_Num
• DR2.name=DR1.name
• DR2.date=CURRENT TIME
• DR2.pre_revision=DR1
• DR2.base_version=DR1.base_version
S506
DR2.File_name=DR2.name+DR2.revision_No+DR2.date
S507
GENERATE NEW INSTANCE RH2 OF Revision_History & STORE IT INTO Revision Tree DATA 18
• RH2.File_name=DV2.File_name+''history''
S508
DR2.has_history = RH2
S509
OBTAIN QUALITY LEVEL L1 OF DICTIONARY SYSTEM FROM DICTIONARY QUALITY DATA 26
S510
GENERATE NEW INSTANCE DQ2 OF DB_quality & STORE IT INTO Revision Tree DATA 18
• DQ2.File_name=DR2.File_name+quality
• DQ2.level=L1
S511
STORE DICTIONARY QUALITY DATA 26 INTO DICTIONARY SYSTEM QUALITY DATA 19 UNDER NAME OF DQ2.File_name
S512
DR2.its_quality = DQ2
S513
STORE BSU CODE CHANGE DATA 9 INTO BSU CODE CHANGE MANAGEMENT EDITING DATABASE 16 (FILE NAME IS RH2.File_name)
S514
STORE ELECTRONIC CATALOG DICTIONARY DATA 10 INTO ELECTRONIC CATALOG DICTIONARY EDITING DATABASE 14 (FILE NAME IS DR2.File_name)
S515
RETURN

FIG. 23

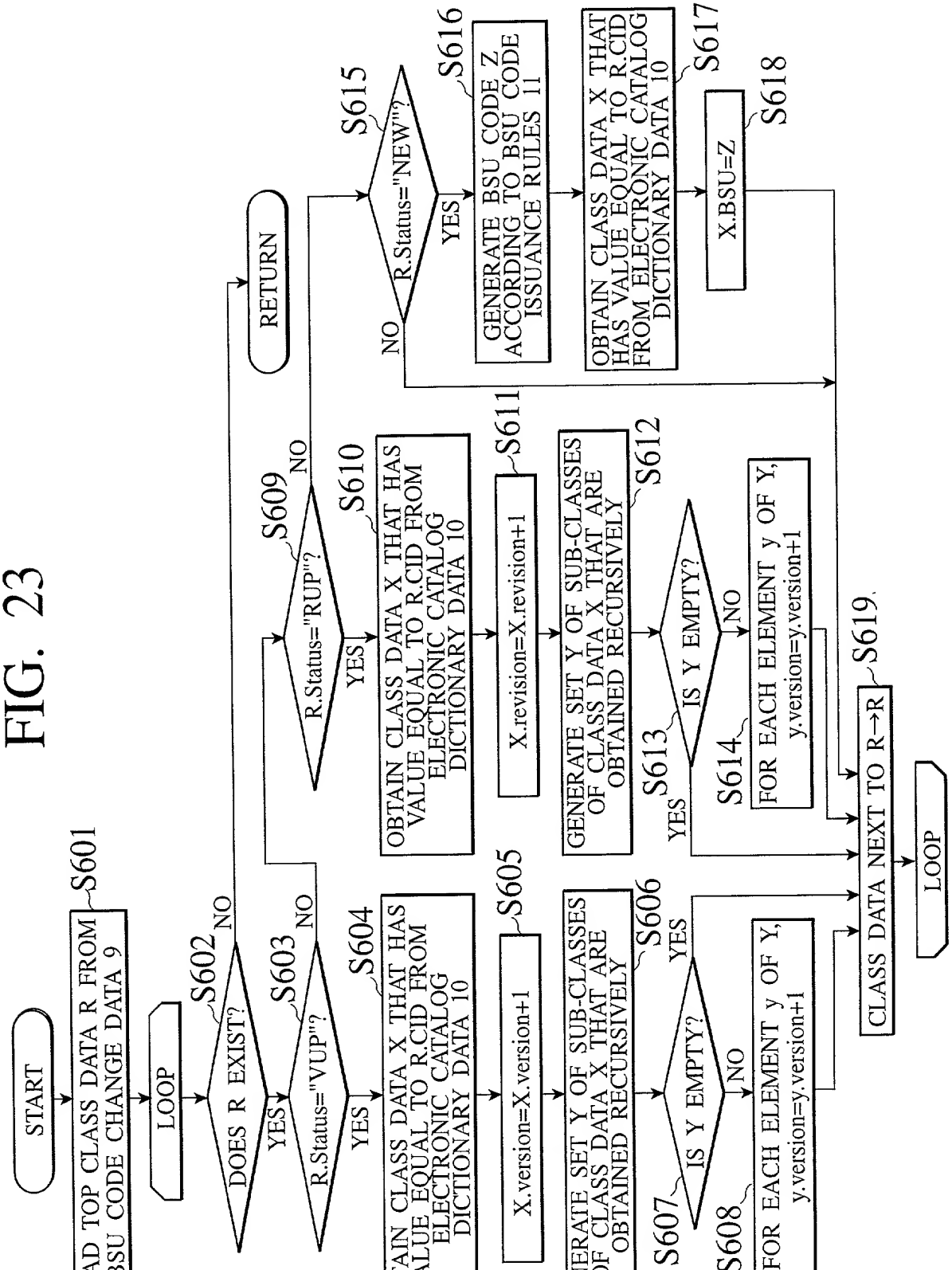

START
READ TOP CLASS DATA R FROM BSU CODE CHANGE DATA 9
S601
LOOP
DOES R EXIST?
S602
NO
YES
R.Status="VUP"?
S603
NO
YES
OBTAIN CLASS DATA X THAT HAS VALUE EQUAL TO R.CID FROM ELECTRONIC CATALOG DICTIONARY DATA 10
S604
X.version=X.version+1
S605
GENERATE SET Y OF SUB-CLASSES OF CLASS DATA X THAT ARE OBTAINED RECURSIVELY
S606
IS Y EMPTY?
S607
YES
NO
FOR EACH ELEMENT y OF Y, y.version=y.version+1
S608
CLASS DATA NEXT TO R→R
S619
LOOP
RETURN
R.Status="RUP"?
S609
NO
YES
OBTAIN CLASS DATA X THAT HAS VALUE EQUAL TO R.CID FROM ELECTRONIC CATALOG DICTIONARY DATA 10
S610
X.revision=X.revision+1
S611
GENERATE SET Y OF SUB-CLASSES OF CLASS DATA X THAT ARE OBTAINED RECURSIVELY
S612
IS Y EMPTY?
S613
YES
NO
FOR EACH ELEMENT y OF Y, y.version=y.version+1
S614
R.Status="NEW"?
S615
NO
YES
GENERATE BSU CODE Z ACCORDING TO BSU CODE ISSUANCE RULES 11
S616
OBTAIN CLASS DATA X THAT HAS VALUE EQUAL TO R.CID FROM ELECTRONIC CATALOG DICTIONARY DATA 10
S617
X.BSU=Z
S618

FIG. 24

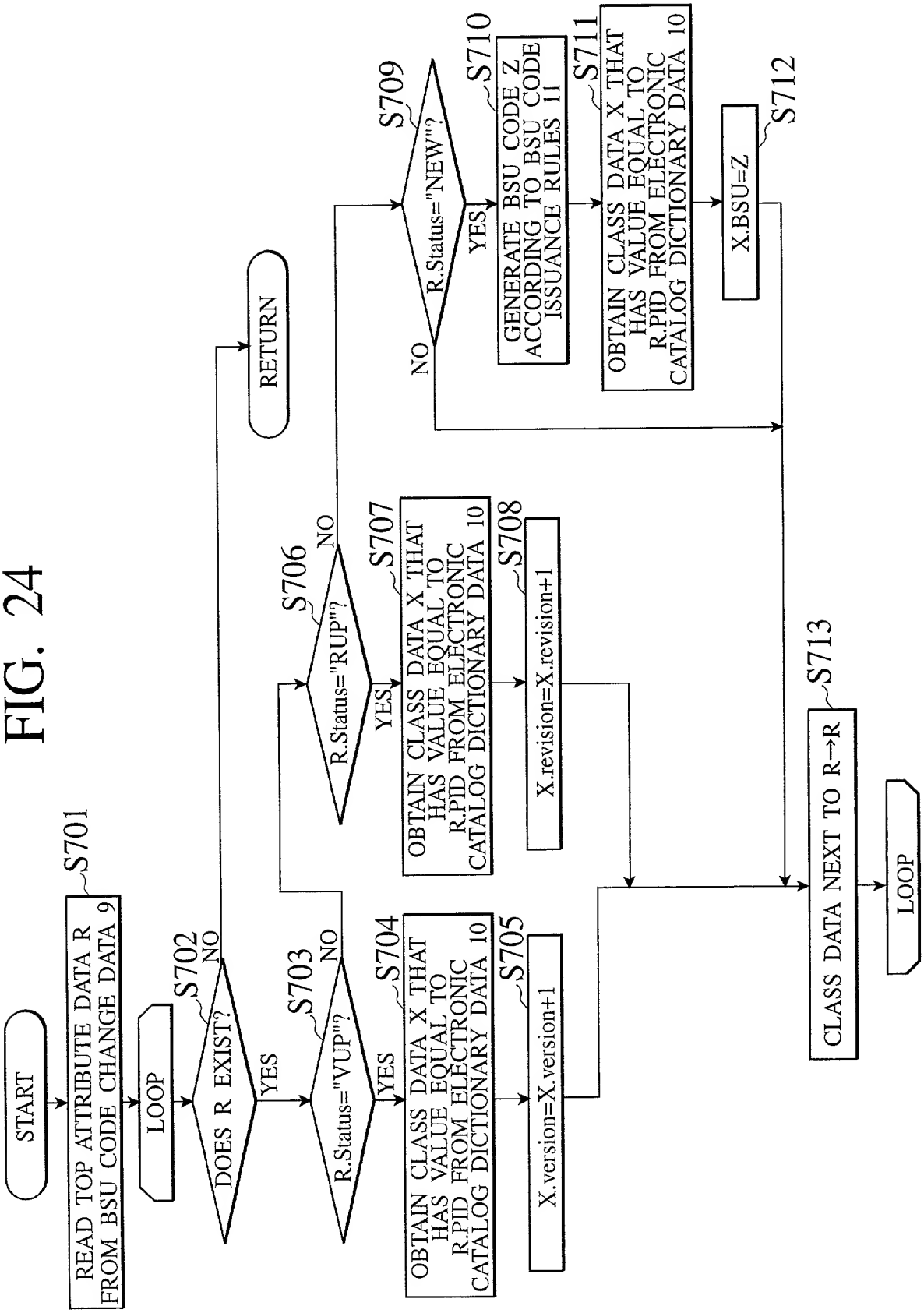

START
READ TOP ATTRIBUTE DATA R FROM BSU CODE CHANGE DATA 9
S701
LOOP
DOES R EXIST?
S702
NO
YES
RETURN
R.Status="VUP"?
S703
NO
YES
OBTAIN CLASS DATA X THAT HAS VALUE EQUAL TO R.PID FROM ELECTRONIC CATALOG DICTIONARY DATA 10
S704
X.version=X.version+1
S705
R.Status="RUP"?
S706
NO
YES
OBTAIN CLASS DATA X THAT HAS VALUE EQUAL TO R.PID FROM ELECTRONIC CATALOG DICTIONARY DATA 10
S707
X.revision=X.revision+1
S708
R.Status="NEW"?
S709
NO
YES
GENERATE BSU CODE Z ACCORDING TO BSU CODE ISSUANCE RULES 11
S710
OBTAIN CLASS DATA X THAT HAS VALUE EQUAL TO R.PID FROM ELECTRONIC CATALOG DICTIONARY DATA 10
S711
X.BSU=Z
S712
CLASS DATA NEXT TO R→R
S713
LOOP

FIG. 25

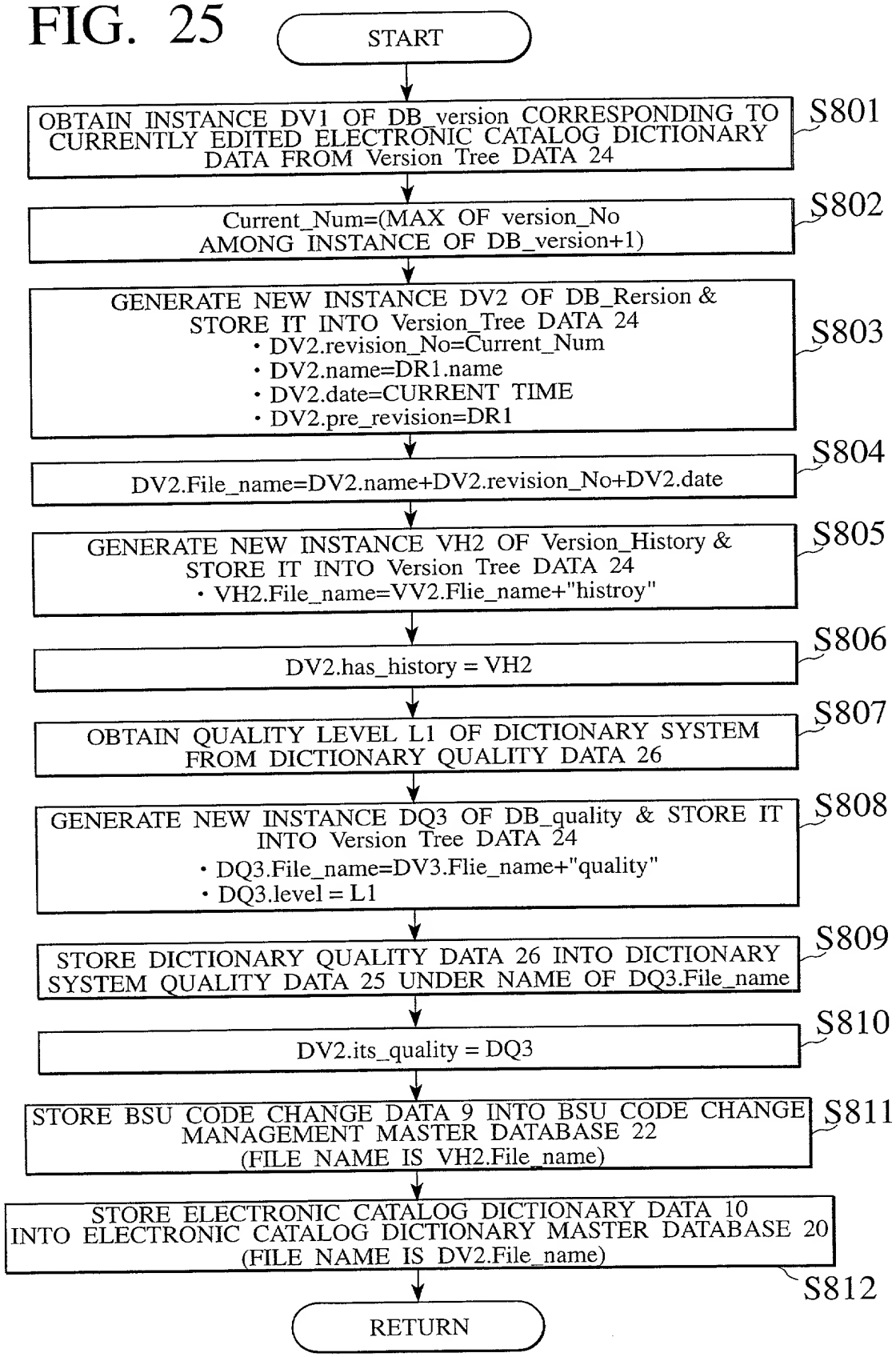

START
OBTAIN INSTANCE DV1 OF DB_version CORRESPONDING TO CURRENTLY EDITED ELECTRONIC CATALOG DICTIONARY DATA FROM Version Tree DATA 24
S801
Current_Num=(MAX OF version_No AMONG INSTANCE OF DB_version+1)
S802
GENERATE NEW INSTANCE DV2 OF DB_Rersion & STORE IT INTO Version_Tree DATA 24
• DV2.revision_No=Current_Num
• DV2.name=DR1.name
• DV2.date=CURRENT TIME
• DV2.pre_revision=DR1
S803
DV2.File_name=DV2.name+DV2.revision_No+DV2.date
S804
GENERATE NEW INSTANCE VH2 OF Version_History & STORE IT INTO Version Tree DATA 24
• VH2.File_name=VV2.Flie_name+"histroy"
S805
DV2.has_history = VH2
S806
OBTAIN QUALITY LEVEL L1 OF DICTIONARY SYSTEM FROM DICTIONARY QUALITY DATA 26
S807
GENERATE NEW INSTANCE DQ3 OF DB_quality & STORE IT INTO Version Tree DATA 24
• DQ3.File_name=DV3.Flie_name+"quality"
• DQ3.level = L1
S808
STORE DICTIONARY QUALITY DATA 26 INTO DICTIONARY SYSTEM QUALITY DATA 25 UNDER NAME OF DQ3.File_name
S809
DV2.its_quality = DQ3
S810
STORE BSU CODE CHANGE DATA 9 INTO BSU CODE CHANGE MANAGEMENT MASTER DATABASE 22 (FILE NAME IS VH2.File_name)
S811
STORE ELECTRONIC CATALOG DICTIONARY DATA 10 INTO ELECTRONIC CATALOG DICTIONARY MASTER DATABASE 20 (FILE NAME IS DV2.File_name)
S812
RETURN

FIG. 26

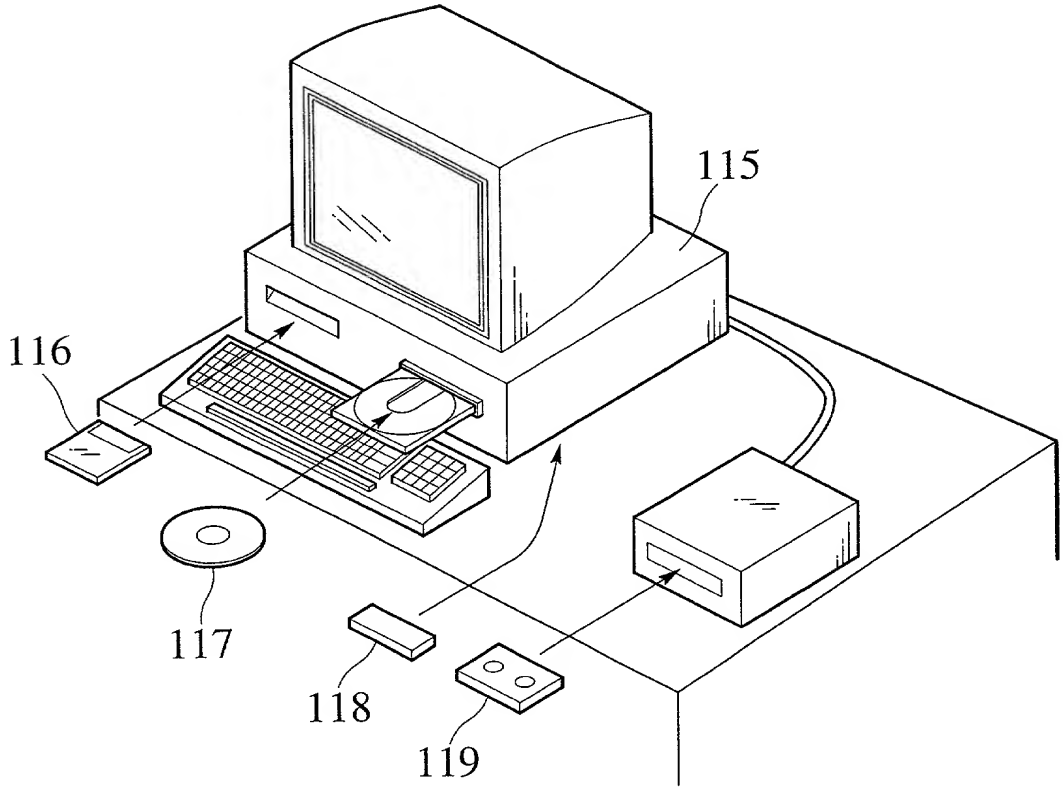
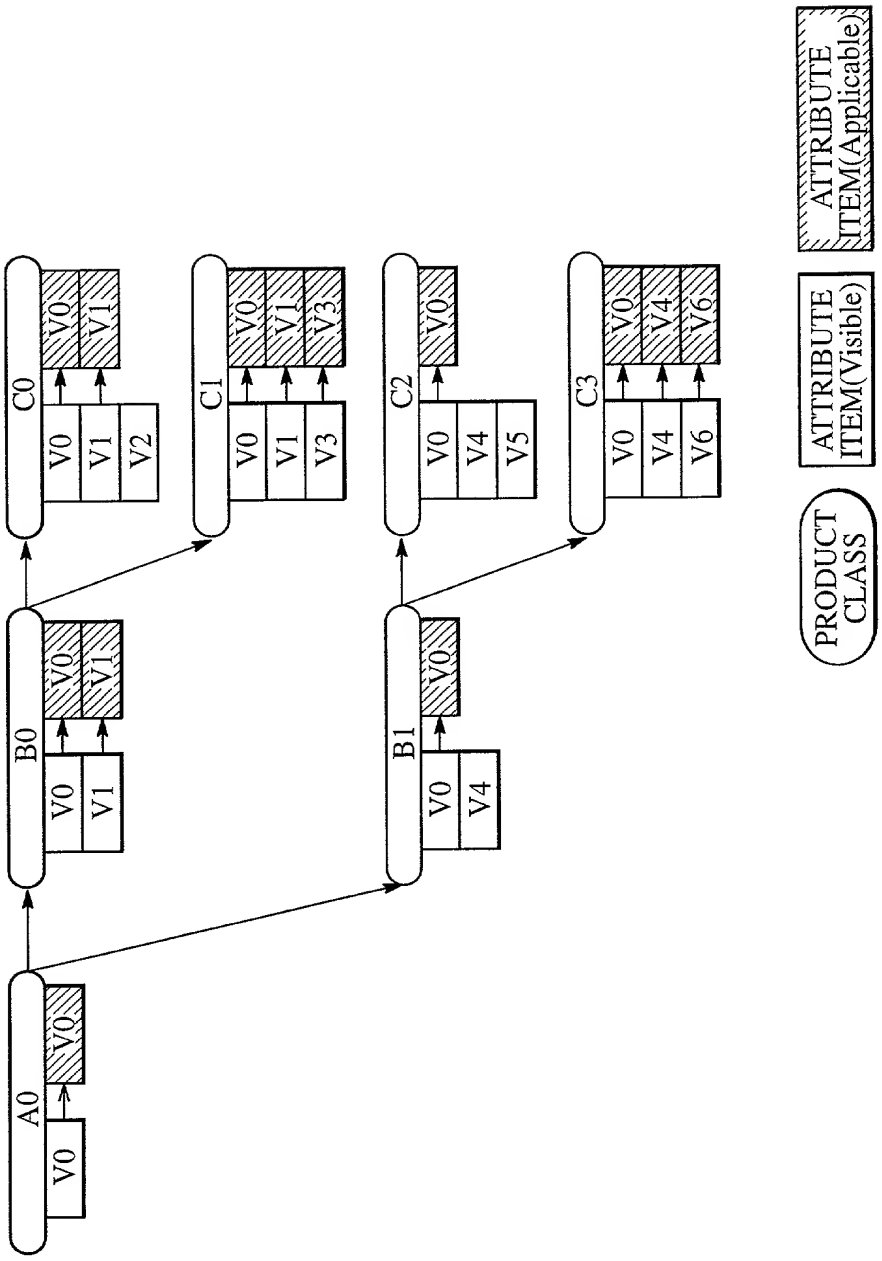


FIG. 27
PRIOR ART



# FIG. 28A

Regarding Rev/Ver of Property

| Concept model | i Data model | Add | Change | Delete |
|---|---|---|---|---|
| Code | Property_BSU.code | — | X | X |
| Definition Class | Property_BSU.name_scope | X | X | X |
| Data_type | Property_DET.domain | — | V(※8) | X |
| Preferred Name | item_name.name | — | R | X |
| Short Name | item_name.Short_name | — | R | X |
| Preferred Letter Symbol | Property_DET.preferred_symbol | R | R | X |
| Synonymous Letter Symbol | Property_DET.synonymous_symbols | R | R | R |
| Synonymous Name | item_name.Synonymous_names | R | R | R |
| Property Type Classification | Property_DET.DET_classification | — | R(※9) | X |
| Definition | Property_DET.definition | — | R/V(※10) | X |
| Source Document of definition | identified_document.document_identifier | R(※11) | R(※11) | R |
| Note | Property_DET.note | R(※11) | R(※11) | R |
| Remark | Property_DET.remark | R(※11) | R(※11) | R |
| Unit | int/real_measure_type.unit.structured_representation | — | X | X |
| Condition | Property_DET.depends_on | X | X/R(※12) | X |
| Formula | Property_DET.formula | R(※11) | R(※11) | R(※11) |
| Format | Property_DET.domain.simple_type.value_format | — | V | X |
| Date of Original Definition | dates.date_of_original_definition | — | X | X |
| Date of current version | dates.date_of_current_version | — | V | X |
| Date of current revision | dates.date_of_current_revision | — | R/V | X |
| Version number | Property_BSU.version | — | V | X |
| Revision number | Property_DET.revision | — | R | X |

※8 Change of Data_type itself not allowed. Possible are:
1. Addition of option of enumeration type data
2. Change of Version of class constituting attribute range(ex:is_part_of relationship)
3.Change of Version of Data_type BSU

※9 Only in case of Rev change of ISO31 or ISO13584.

※10 Change of definition that affects Property is not allowed. But processing for evaluating value of this Property may be Changed. Ver change in such case, or Rev change otherwise.

※11 Change of definition that affects Property is not allowed.

※12 Change of Condition_DET is not allowed. Version change of Condition_DET will propagate to Depend_P_DET.

OBLON, SPIVAK, ET AL
DOCKET #: 200023US2RD
INV: Satoshi ITO
SHEET 25 OF 34
25/34

# FIG. 28B

Regarding Rev/Ver of Class

| Concept model | i Data model | Add | Change | Delete |
|---|---|---|---|---|
| Code | Class_BSU.code | — | X | X |
| Super Class | Class.its_superclass | V(※15) | V(※12) | V(※12) |
| Preferred Name | item_name.name | — | R | X |
| Short Name | item_name.Short_name | V | V(※16) | X |
| Visible Type | Class_BSU.added_visible_data_type | V | V(※16) | X |
| Applicable Type | Class.defined_types | V | V(※16) | X |
| Sub-class selection Property | class.sub_class_properties | R | R | R |
| Synonymous Name | item_name.Synonymous_names | V | V(※16) | X |
| Visible Property | Class_BSU.added_visible_property | V | V(※16) | X |
| Applicable Property | Described_by | V | V(※16) | X |
| Class Value assignment | Class.class_constant_values | — | R/V(※17) | X |
| Definition | Property_DET.definition | R | R | R |
| Source Document of definition | identified_document.document_identifier | R | R | R |
| Note | Property_DET.note | R | R | R |
| Remark | Property_DET.remark | — | R | R |
| Simplified Drawing | Class.simplified_drawing | — | X | X |
| Date of Original Definition | dates.date_of_original_definition | — | V | X |
| Date of current version | dates.date_of_current_version | — | R/V | X |
| Date of current revision | dates.date_of_current_revision | — | V | X |
| Version number | Property_BSU.version | — | R | X |
| Revision number | Property_BSU.revision | — | R | X |

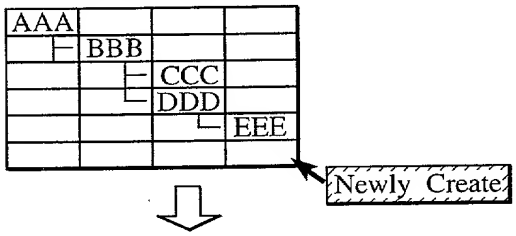
※13 Value can only be added.
※14 Change of arbitrary element of Dependent_P_DET Set.
※15 Property /Data_type of inherited Visible/Applicable Should not be deleted by change of Super Class. This occurs when (1) new Version is created by Sub-Class, or(2) intermediate class is inserted.
※16 Change is only addition of new Item or change of Version of Item to be referred.
※17 Change of definition that affects Property is not allowed.

Propagation of change of Class V/R

- Version change of parent class→Version charge of all children classes (grand children classes)
- Version change of Table or contents that refer to class→Version change of that class

# FIG. 29
PRIOR ART

CREATION OF END CLASS

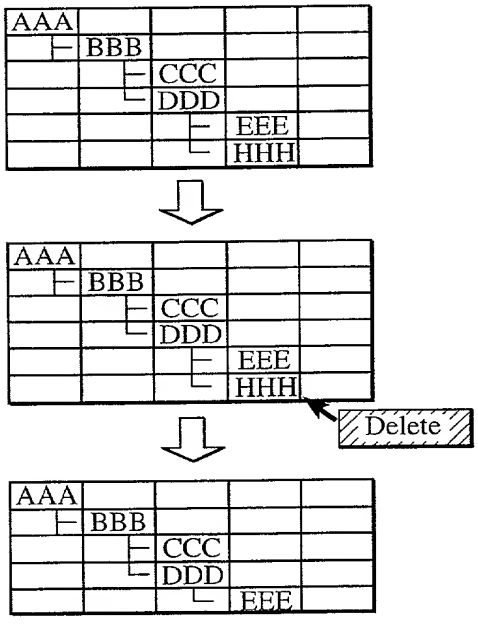



| Class Name : FFF | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| ZZZ | → | | DDD |

Class HHH → NewBSU

# FIG. 30
PRIOR ART

DELETION OF END CLASS

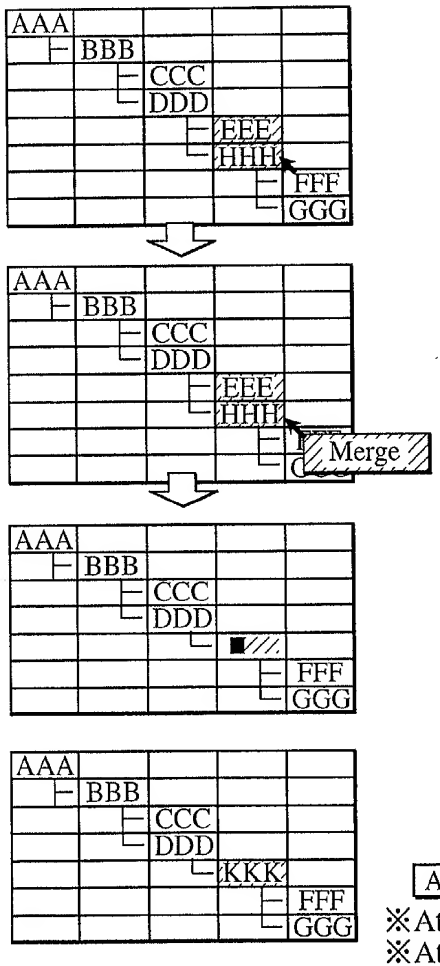



| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| ZZZ | → | | DDD |
| SSS | → | | |

Old HHH → Retired
Old SSS → Retired

# FIG. 31 PRIOR ART

MERGING OF CLASSES




| Class Name : EEE | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| ZZZ | → | ZZZ | |

| CLASS NAME : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| XXX | → | | |

| CLASS NAME : FFF | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| XXX | → | | HHH |
| SSS | → | | |

Class EEE → Retired
Class HHH → Retired

Class Name : EEE → : Vanish
Class Name : HHH → : Vanish

OR TYPE MERGE

| Class Name : KKK | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| ZZZ | → | ZZZ | |
| XXX | → | | |

| CLASS NAME : FFF | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| ZZZ | → | ZZZ | KKK |
| XXX | → | | KKK |
| SSS | → | | |

Class KKK → NewBSU
Attribute ZZZ → NewBSU※4
Attribute XXX → NewBSU※4
Old ZZZ → Retired
Old XXX → Retired

Class FFF → NewBSU※3&2
Old FFF → Retired
Attribute SSS → NewBSU※4
Old SSS → Retired

AND TYPE MERGE

| Class Name : KKK | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |

| CLASS NAME : FFF | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| XXX | → | | |
| SSS | → | | |

Class KKK → NewBSU

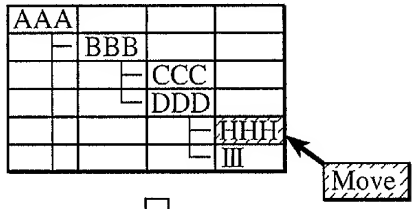
Class FFF → NewBSU※3&2
Attribute XXX → NewBSU※4
Attribute SSS → NewBSU※4
Old ZZZ → Retired
Old XXX → Retired
Old SSS → Retired

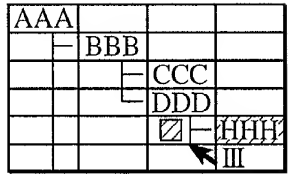
Attribute Name : ZZZ → : Vanish

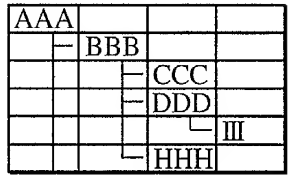
※Attribute SSS of HHH is inherited to lower level
※Attribute SSS of EEE is vanished as there is no one to inherit
※Among attributes common to classes to be merged:
1: an attribute is Applicable in new class if it is Applicable in all classes
2: an attribute is Visible in new class if it is Visible in at least one class
※Among attributes not common to classes to be merged:
1: an attribute is Visible in new class

# FIG. 32
PRIOR ART

MOVING OF CLASS (1)




| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| YYY | → | | DDD |
| ZZZ | → | | |

| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| YYY | → | | |
| ZZZ | → | | |

※YYY & DDD are overlapping

Class HHH → NewBSU※3&2
Old HHH → Retired

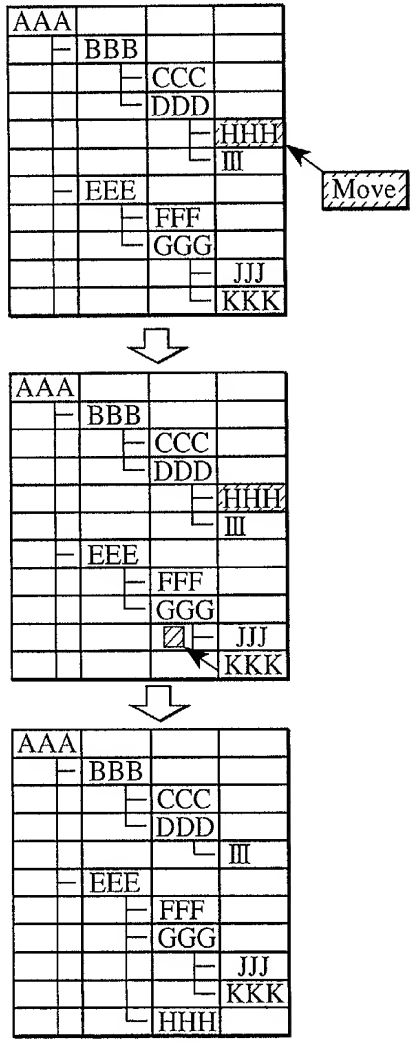
Attribute YYY → NewBSU※4
Attribute ZZZ → NewBSU※4
Old ZZZ → Retired

※Old YYY is used in Class III

# FIG. 33 PRIOR ART

MOVING OF CLASS (2)

Example 2




| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| YYY | → | | DDD |
| ZZZ | → | | |

| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| TTT | × | TTT | EEE |
| SSS | → | | |
| YYY | → | | |
| ZZZ | → | | |

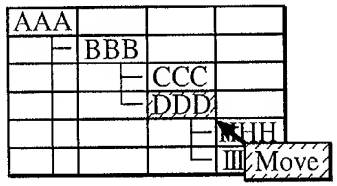
Class HHH → NewBSU※3
Old HHH → Retired
Attribute SSS → NewBSU※4
Attribute YYY → NewBSU※4
Attribute ZZZ → NewBSU※4
Old ZZZ → Retired
※Old SSS is used in Class DDD
※Old YYY is used in Class III

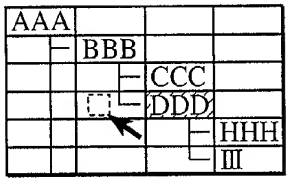
※Inherited attribute is changed to direct attribute
※New inherited attribute is added
※SSS & YYY are overlapping → Assist Function

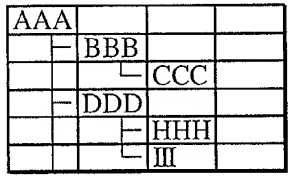
# FIG. 34
PRIOR ART

MOVING OF CLASS (3)

Example 3




| Class Name : DDD | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| ZZZ | → | | |

| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| ZZZ | → | | DDD |
| XXX | → | | |

| Class Name : DDD | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | |
| ZZZ | → | | |

| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | DDD |
| ZZZ | → | | DDD |
| XXX | → | | |

※Inherited attribute is changed to direct attribute

Class DDD → New BSU※3&2
Old DDD → Retired
Attribute SSS → New BSU※4
※Old SSS is used in Class BBB
Attribute ZZZ → New BSU※4
Old ZZZ → Retired

Class HHH → New BSU※2
Old HHH → Retired
Attribute XXX → New BSU※4
Old XXX → Retired

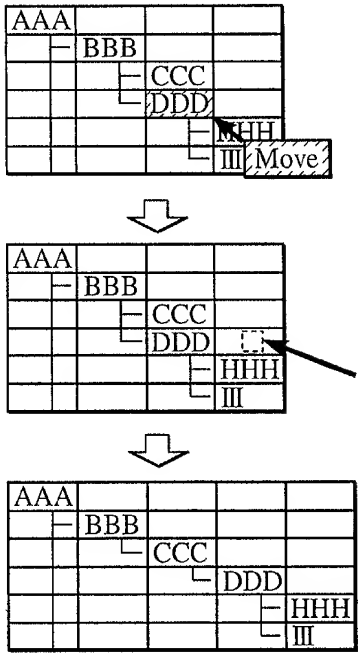
⇩

Similarly for Class III

# FIG. 35
PRIOR ART

MOVING OF CLASS (4)

Example 4




| Class Name : DDD | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| ZZZ | → | | |

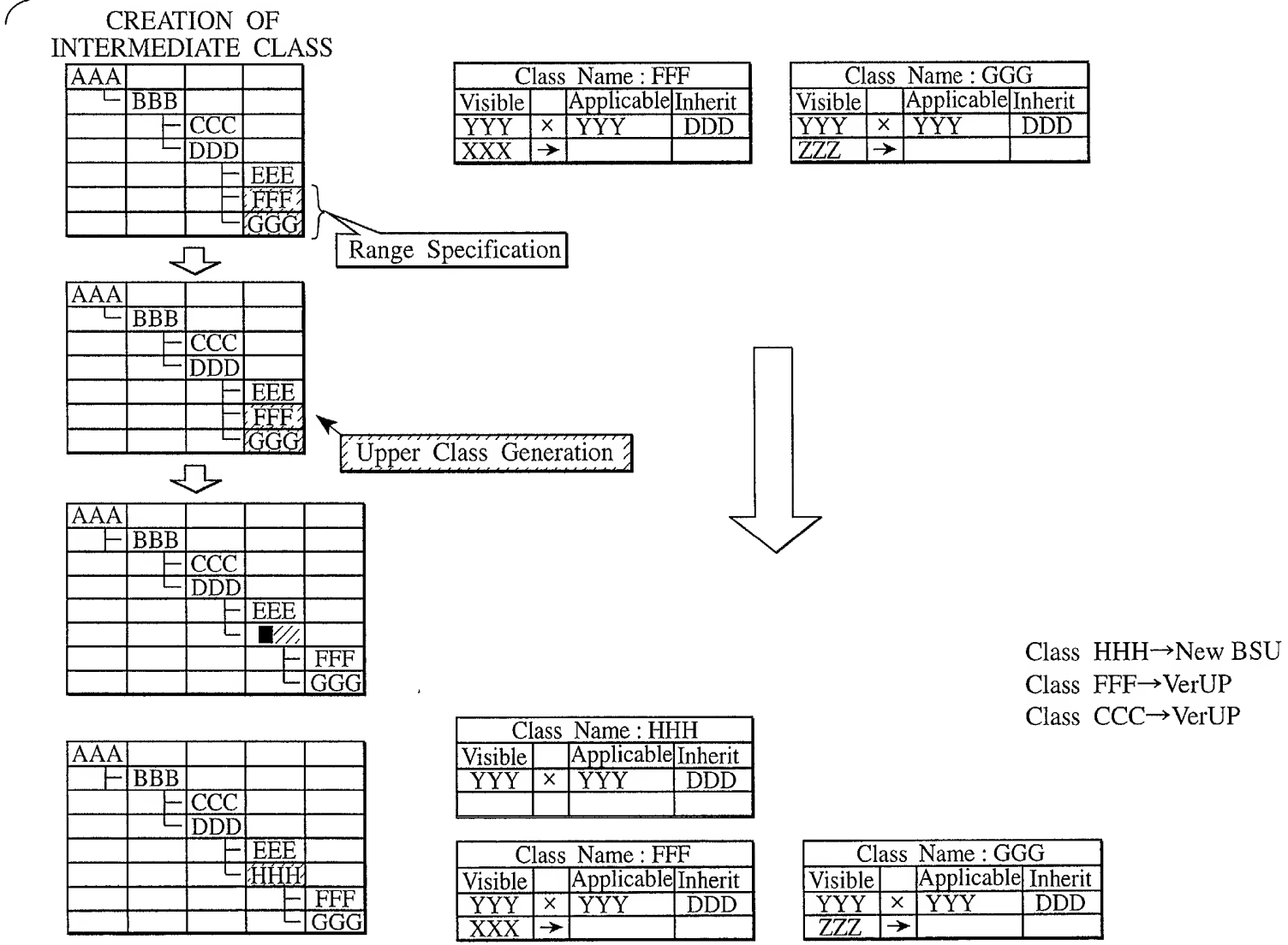
| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| ZZZ | → | | DDD |
| XXX | → | | |

| Class Name : DDD | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| YYY | → | | CCC |
| ZZZ | → | | |

| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| SSS | × | SSS | BBB |
| YYY | → | | CCC |
| ZZZ | → | | DDD |
| XXX | → | | |

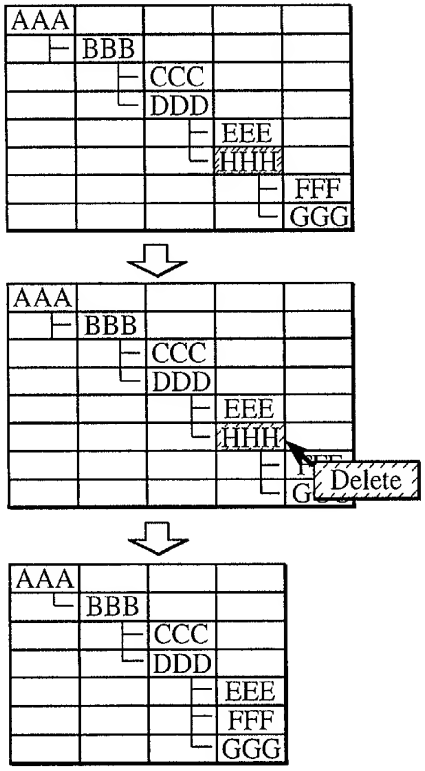
Class DDD → VerUP
Class HHH → VerUP
Class III → VerUP

# FIG. 36 PRIOR ART



# FIG. 37
PRIOR ART

DELETION OF
INTERMEDIATE CLASS




| Class Name : HHH | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| SSS | → | | |

| CLASS NAME : FFF | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| SSS | → | | HHH |
| XXX | → | | |

| CLASS NAME : GGG | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| SSS | → | | HHH |
| ZZZ | → | | |

| CLASS NAME : FFF | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| SSS | → | | |
| XXX | → | | |

| CLASS NAME : GGG | | | |
|---|---|---|---|
| Visible | | Applicable | Inherit |
| YYY | × | YYY | DDD |
| SSS | → | | |
| ZZZ | → | | |

※SSS is overlapping → Assist Function

3G 3120
5756-525-US

# Declaration and Power of Attorney For Patent Application

## 特許出願宣言書及び委任状

## Japanese Language Declaration

## 日本語宣言書

下記の氏名の発明者として、私は以下の通り宣言します。

私の住所、私書箱、国籍は下記の私の氏名の後に記載された通りです。

下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者（下記の名称が複数の場合）であると信じています。

_____

_____

_____

上記発明の明細書は、

☐ 本書に添付されています。

☐ ___月___日に提出され、米国出願番号または特許協定条約国際出願番号を_____とし、
（該当する場合）_____に訂正されました。

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

私は、連邦規則法典第37編第1条56項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled.

ELECTRONIC CATALOG MAINTENANCE SYSTEM FOR ENABLING OUT-OF-STANDARD ELECTRONIC CATALOG CHANGES

the specification of which

☒ is attached hereto.

☐ was filed on _____
as United States Application Number or
PCT International Application Number
_____ and was amended on
_____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

# Japanese Language Declaration
# (日本語宣言書)

私は、米国法典第35編119条（a）-（d）項又は365条（b）項に基づき下記の、米国以外の国の少なくとも一ヵ国を指定している特許協力条約365（a）項に基づく国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している、本出願の前に出願された特許または発明者証の外国出願を以下に、枠内をマークすることで、示しています。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)
外国での先行出願

| (Number) (番号) | (Country) (国名) | (Day/Month/Year Filed) (出願年月日) | Priority Claimed 優先権主張 Yes はい | No いいえ |
|---|---|---|---|---|
| P 11-332038 | Japan | 22/November/1999 | ☒ | ☐ |
| | | | ☐ | ☐ |

私は、第35編米国法典119条（e）項に基づいて下記の米国特許出願規定に記載された権利をここに主張いたします。

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

| (Application No.) (出願番号) | (Filing Date) (出願日) |
|---|---|
| | |

私は、下記の米国法典第35編120条に基づいて下記の米国特許出願に記載された権利、又は米国を指定している特許協力条約365条（c）に基づく権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第35編112条第1項又は特許協力条約で規定された方法で先行する米国特許出願に開示されていない限り、その先行米国出願書提出日以降で本出願書の日本国内または特許協力条約国際提出日までの期間中に入手された、連邦規則法典第37編1条56項で定義された特許資格の有無に関する重要な情報について開示義務があることを認識しています。

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of application.

| (Application No.) (出願番号) | (Filing Date) (出願日) | (Status: Patented, Pending, Abandoned) (現況：特許許可済、係属中、放棄済) |
|---|---|---|
| | | |
| | | |

私は、私自信の知識に基づいて本宣言書中で私が行なう表明が真実であり、かつ私の入手した情報と私の信じるところに基づく表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第18編第1001条に基づき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行なえば、出願した、又は既に許可された特許の有効性が失われることを認識し、よってここに上記のごとく宣誓を致します。

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

# Japanese Language Declaration
# (日本語宣言書)

委任状：私は下記の発明者として、本出願に関する一切の手続きを米特許商標局に対して遂行する弁理士または代理人として、下記の者を指名いたします。
(弁護士、または代理人の指名及び登録番号を明記のこと)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith: *(list name and registration number)*

Norman F. Oblon, Reg. No. 24,618; Marvin J. Spivak, Reg. No. 24,913; C. Irvin McClelland, Reg. No. 21,124; Gregory J. Maier, Reg. No. 25,599; Arthur I. Neustadt, Reg. No. 24,854; Richard D. Kelly, Reg. No. 27,757; James D. Hamilton, Reg. No. 28,421; Eckhard H. Kuesters, Reg. No. 28,870; Robert T. Pous, Reg. No. 29,099; Charles L. Gholz, Reg. No. 26,395; William E. Beaumont, Reg. No. 30,996; Robert F. Gnuse, Reg. No. 27,295; Jean-Paul Lavalleye, Reg. No. 31,451; Stephen G. Baxter, Reg. No. 32,884; Richard L. Treanor, Reg. No. 36,379; Steven P. Weihrouch, Reg. No. 32,829; John T. Goolkasian, Reg. No. 26, 142; Richard L. Chinn, Reg. No. 34,305; Steven E. Lipman, Reg. No. 30,011; Carl E. Schlier, Reg. No. 34,426; James J. Kulbaski, Reg. No. 34,648; Richard A. Neifeld, Reg. No. 35,299; J. Derek Mason, Reg. No. 35,270; Surinder Sachar, Reg. No. 34,423; Christina M. Gadiano, Reg. No. 37,628; Jeffrey B. McIntyre, Reg. No. 36,867; William T. Enos, Reg. No. 33,128 and Michael E. McCabe, Jr., Reg. No. 37,182, with full powers of substitution and revocation.

書類送付先

Send Correspondence to:

OBLON, SPIVAK, McCLELLAND, MAIER & NEUSTADT, P.C.
FOURTH FLOOR
1755 JEFFERSON DAVIS HIGHWAY
ARLINGTON, VIRGINIA 22202 U.S.A.

直接電話連絡先：（名前及び電話番号）

Direct Telephone Calls to: (name and telephone number)

(703) 413-3000

| | |
|---|---|
| 単独発明者または第一の共同発明者の氏名 | Full name of sole or first joint inventor<br>Satoshi ITO |
| 発明者の署名　日付 | Inventor's signature Satoshi Ito Date November 17, 2000 |
| 住所 | Residence<br>Tokyo, Japan |
| 国籍 | Citizenship<br>Japan |
| 郵便の宛先 | Post Office Address<br>c/o Intellectual Property Division, |
| | Toshiba Corporation, 1-1-1, Shibaura, Minato-ku, Tokyo, Japan |
| 第二の共同発明者の氏名 | Full name of second joint inventor, if any |
| 第二の共同発明者の署名　日付 | Second joint Inventor's signature Date |
| 住所 | Residence |
| 国籍 | Citizenship |
| 郵便の宛先 | Post Office Address |
| | |

(第三以降の共同発明者についても同様に記載し、署名すること)

(Supply similar information and signature for third and subsequent joint inventors.)